



DEPLOYMENT GUIDE

MOBILE TICKET

224.06B

Copyright notice

The information in this document is subject to change without prior notice and does not represent a commitment on the part of Q-MATIC AB.

All efforts have been made to ensure the accuracy of this manual, but Q-MATIC AB can not assume any responsibility for any errors and their consequences.

This manual is copyrighted and all rights are reserved.

Reproduction of any part of this manual, in any form, is not allowed, unless written permission is given by Q-MATIC AB.

COPYRIGHT © Q-MATIC AB, 2025.

ⓘ Please send any feedback or questions about this manual to documentation@qmatic.com.

Contents

1. What's new.....	1
2. Introduction	3
2.1. Overview	4
2.2. System requirements	5
2.3. Architecture	6
2.4. Limitations	7
2.5. Browsers and cookies	7
2.6. Software deliverables	8
3. Requirements and Preparations ...	11
3.1. Orchestra preparations	12
3.2. URLs for notifications and QR codes	14
4. Installation API Gateway.....	17
4.1. Install API Gateway	18
4.2. Configure API Gateway for Mobile Ticket	19
5. Installation and Upgrade	21
5.1. Install Mobile Ticket	22
5.2. Enable HTTPS / SSL	24
5.3. Install Mobile Ticket as a service	27
5.4. Database configuration	28
5.5. Upgrade Mobile Ticket	30
6. Configuration.....	31
6.1. About configuration of Mobile Ticket	32
6.2. Change which information is displayed	32
6.3. Change functional settings	35
6.4. Change appointment settings	39
6.5. Allow personal information	40
6.6. Configure open hours	42
6.7. Localize Mobile Ticket	44
6.8. Manage notification sound	46
6.9. Enable Google Analytics	46
6.10. Configure security settings	47
6.11. Manage background time intervals	48
7. Branding.....	49

- 7.1. Introduction50
- 7.2. Accessibility guidelines52
- 7.3. Color reference information52
- 7.4. Customize colors56
- 7.5. Customize logo57
- 8. Troubleshooting 59**
 - 8.1. Troubleshooting guide60
 - 8.2. Frequently Asked Questions62
- 9. Walk-through 63**
 - 9.1. Use cases64
 - 9.2. End user walk-through65

1. What's new

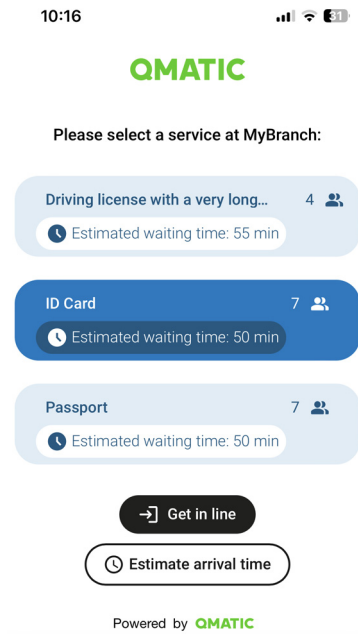
	Chapter	Change
05.A	• Installation and configuration	• Branding moved to new chapter • Changes to Additional data parameter
	• Upgrade	• New chapter for upgrade
05.B	• Branding	• More customization options added
05.C	• Installation and configuration	• Security and HTTPS configuration parameter added
05.D	• Installation and configuration	• Enable visit delay re-added
05.E	• Installation and configuration	• Added branch selection URL and Mobile Ticket URL to Enable additional data
05.F	• Installation and configuration	• first_name and last_name parameters added
05.G	• Installation and configuration	• Reschedule link and dynamic URL added
05.I	• Installation and configuration	• Language picker added
05.J	• Installation and configuration	• Configuration option to enable message box with link to form or website added

	Chapter	Change
05.K	• Installation and configuration	• URL with external ID added as appointment check-in URL • Configuration option to display custom fields below ticket added • Configuration option to show QR code in order to identify the visit added. • Configuration option to set number of queue poll retry attempts added • Configuration option to show accessibility statement added
	• Requirements and preparation	• Removed references to Teams as communication platform for remote service delivery as there are other supported communication platforms.
05.L	• Installation and Configuration	• Information about display language added in Localization section.
06.A	• General	• Updated whole manual to new template • Changed the structure of some chapters
	• Configuration	• Added section about estimated waiting time configuration.
06.B	• Configuration	• Add settings for polling intervals and retry attempts.

2. Introduction

Overview	4
System requirements	5
Architecture	6
Limitations	7
Browsers and cookies	7
Software deliverables	8

2.1 Overview



With the Mobile Ticket application, customers can get in line for a service from their mobile device, via a web page/web application, using a direct URL, or for example a QR code. They can then wait to be called wherever they want, monitoring their position in line on their device.

2.1.1 Main features

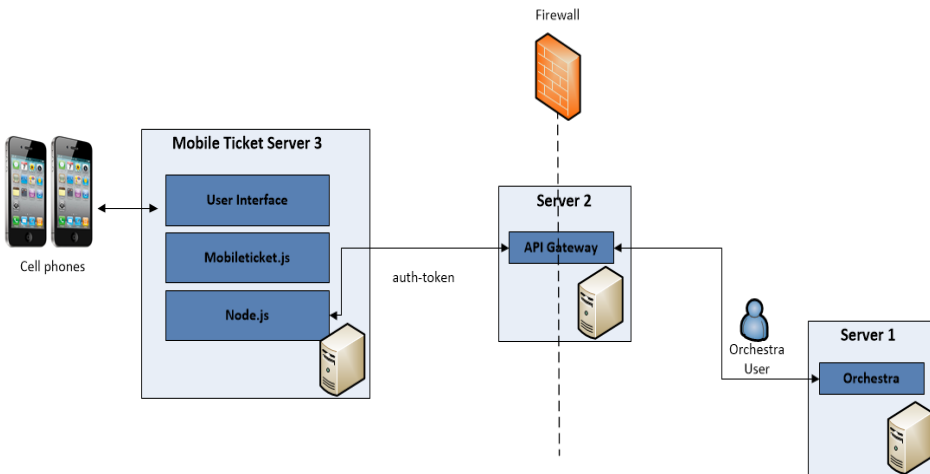
- Seamless entry with QR codes and URLs
- Possibility for customer to enter personal data such as phone number
- Can be combined with SMS notifications and Customer Feedback
- Check-in for appointments
- Can be linked to Google Analytics
- Support for multi-language and rtl
- Branding for colors and logo
- One-time password and other security features that prevent misuse of system and hacker attacks
- Integrations with communication platforms for remote serving
- Estimated waiting time

2.2 System requirements

- Orchestra 7.0 update 12, or later, is required for Mobile Ticket. However, some features may require a certain version of a module, see below.
- If you want customers to be able to add their phone number to the visit, you need Notification Admin 4.0.0.49 or later.
- For remote service delivery, Counter and Qmatic Web Booking need to be installed. For more information, see the configuration guides for Remote Service.
- If service groups are enabled, General Admin application is required.
- For translations of service and branch names, General Admin 1.6.1 is required.
- HTTPS is needed in order to use location detection (and list branches based on user's position).
- For branches to be grayed out when there are no available services, Orchestra build 4.4.0.290 is required (this applies when Service Scheduler is used).

2.3 Architecture

The picture below illustrates the architecture of the Mobile Ticket application, as well as how the different parts communicate with each other:



2.4 Limitations

- When mobile user credentials are needed, *Superadmin* user can not be used.
- It is not supported to create multi-service visits in Mobile Ticket. However, it is possible to for example create a multi-service visit in a kiosk, or with Qmatic Web Booking, and then follow the multi-service visit in Mobile Ticket.
- When using distributed operations and Mobile Ticket, stable connectivity to Central is required.

2.5 Browsers and cookies

2.5.1 Supported Browsers

Mobile Ticket is tested and supported on the following browsers:

- Safari
- Chrome
- Opera
- Firefox (PC)
- Edge (PC)

If you use Mobile Ticket on a browser that is not supported, you will be notified about that.

2.5.2 Cookies

In Mobile Ticket, by default, no cookies are stored. Instead local storage is used.

However, there is an option to enable Google Analytics. If so, it might be required to enable cookie consent in Mobile Ticket in order for follow local regulations related to GDPR or similar.

2.6 Software deliverables

Please read the *README.md* file for more information.

The following table describes the contents of the Mobile Ticket package, on a high level:

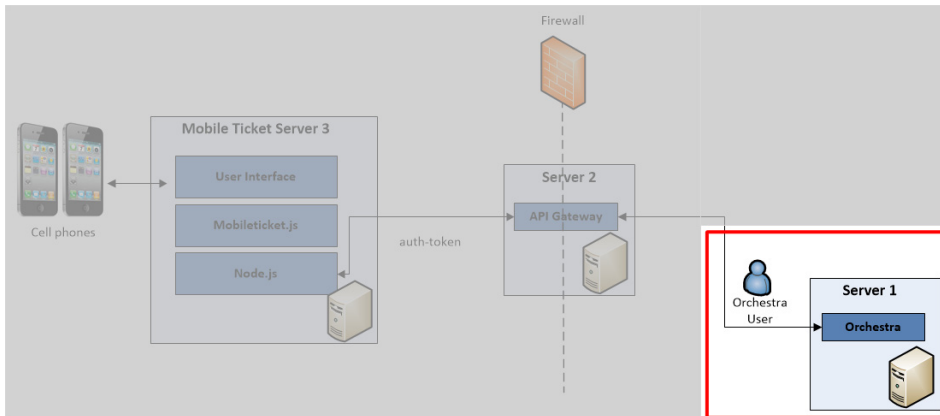
File / folder	Sub folder	Contents Description
sslcrt		This folder contains a script called <i>create_cert.sh</i>. This folder is also where you should place your certificate, if your server is supporting SSL. This is defined in the <i>proxy-config.json</i> file. It also contains the certificate <i>Global.cer</i>.
node_modules	iconv-lite express express-http-proxy raw-body unpipe	These sub folders contain parsers, http-proxy, readme-files etc.
proxy-config.json		This is where you specify the API Gateway Host address, local webserver port, etc. For more information, see “5.1.2 Step 2: Configure the proxy/webserver.” on page 22.
server.js		Main script file to start the solution.

File / folder	Sub folder	Contents Description
src	app	This folder contains the following sub folders: • <i>config</i> - This folder contains the <i>config.json</i> file, where you specify the branch radius, where you find the current version number of Mobile Ticket, and so on. For more information, see “6. Configuration” on page 31. • <i>locale</i> - This folder from start only contains the file <i>en.json</i> and this is where you place your translated language files, if applicable. For more information, see “6.7 Localize Mobile Ticket” on page 44. • <i>resources</i> - In this folder, you find graphical resources used, such as fonts, images, and buttons. A few of these are then pointed out from the <i>theme-styles.css</i> file. Here, you can also find the sound file <i>notification.mp3</i>, which is pointed out from the <i>config.json</i> file.
	zip	This folder contains the files <i>bundle.min.css.gz</i> and <i>bundle.min.js.gz</i>.
	The files <i>favicon.ico</i> and <i>index.html</i>	

3. Requirements and Preparations

Orchestra preparations	12
URLs for notifications and QR codes	14

3.1 Orchestra preparations



To install Mobile Ticket, the following steps need to be completed in Orchestra:

1. In **System Administration > Unit Types**, add the **VisitApp** unit type.
2. In **System Administration > Settings > Parameters**, make sure that the **mobile user** has the wanted user credentials, in the Mobile API (Central) section. Make a note of the password that you decide to use here, since it will be used in later configuration.

ⓘ The default password for the mobile user is *ulan*. It's recommended to change this password.
3. Still in the Parameters page, enter the **Mobile Ticket Base URL** if you are going to use SMS and email notifications together with Mobile Ticket. The URL is used as a variable in Notification Admin and makes it possible to send personal links to Mobile Ticket or to appointment check-in.
4. In **User Management**, create a user called *mobile* with the same credentials as entered in System Administration. This is required if you use any of the features that require special access: appointment check-in, open hours per service or one-time password feature.
 - As from Orchestra 7.4 update 3 you can assign the pre-configured role **Mobile User** to the mobile user which has all required access modules.
 - For Orchestra 7.4 update 2 and earlier, you need to make sure the mobile user is assigned to a role with the following access modules:
 - Service point connector
 - Entry point connector
 - Appointment connector

Make sure the mobile user has access to all branches.

-
5. In **Business Configuration > Operation Profiles**, add the **VisitApp** unit type to the wanted equipment profiles.
 6. In **Business Configuration > Services / Branches**, make sure that the services and branches that you want to be visible in the Mobile Ticket application have the **Mobile enabled** setting enabled:

Mobile enabled 

7. Enter the branch position (latitude and longitude) for all branches.

Latitude	57.637160000000094	Select position
Longitude	12.017959999999979	

Also, make sure that you enter the address of your branches, so that they will be displayed correctly in the branch selection list.

8. After making changes in Business Configuration, save and publish the wanted branches.

3.2 URLs for notifications and QR codes

URLs can be used to:

- launch Mobile Ticket where the user can select a service and possibly a branch (if branch is not predefined in link)
- create a visit for a specific branch and service
- provide a way for users to check in for appointments.

These URLs can either be sent to customers via sms or email, or be turned into QR codes which can then be placed on for example paper tickets or signs.

⚠ Make sure the Mobile Ticket base URL is configured in System Administration parameters.

3.2.1 URL formats

⚠ In the URLs, both http and https can be used.

Table 1: URL formats

Use case	URL format
Launch Mobile Ticket start page.	https://<mobile_ticket_server>:<port>/ticket
Create visit with pre-defined branch and service	https://<mobile_ticket_server>:<port>/branches/{branchId}/services/{serviceId}
Show service list for predefined branch	https://<mobile_ticket_server>:<port>/branches/{branchId}
Show service list for predefined branch and service group	https://<mobile_ticket_server>:<port>/branches/{branchId}/{serviceGroupId}
Go to appointment check-in	Using public ID: https://<mobile_ticket_server>/appointment?appId={{appointmentPublicId}} Using external ID (to be used for Qmatic Experience Cloud appointments): https://<mobile_ticket_server>/appointment?external={{externalId}}

QR codes

A typical example of how QR codes could work, would be a customer browsing a department store, looking at mobile devices. Next to the merchandise there is a

poster saying “If you need help, scan this QR code!”. The customer scans the QR code and automatically gets a mobile ticket with a visit number for the service Mobile Devices.

A QR code could also be placed on the paper ticket. For that, follow these steps:

1. In the Surface Editor application, create a new ticket layout.
2. Drag and drop the barcode component onto the surface.
3. As barcode type, select **QR Code**.
4. In the System information drop down list, select **Mobile Ticket URL**.

For more information, see the Orchestra Administrator's Guide, found on Qmatic World.

3.2.2 Notifications

ⓘ For notifications to work with phone numbers added from Mobile Ticket UI, you need Notification Admin 4.0.0.49, or later.

Notifications can be used with Mobile Ticket in the following ways (for example):

- **Visit create:** If the visit is created using for example Concierge and there is contact information available on the visit/customer, you can use the Visit create notification type to send a link to open this specific ticket.
- **Visit update:** If a customer has taken a paper ticket and wants a mobile ticket instead, he or she can walk up to a staff member who can find the visit in the queue and then add a phone number to the visit and send SMS.

ⓘ For this functionality to work, Orchestra 7.1 or later is required.

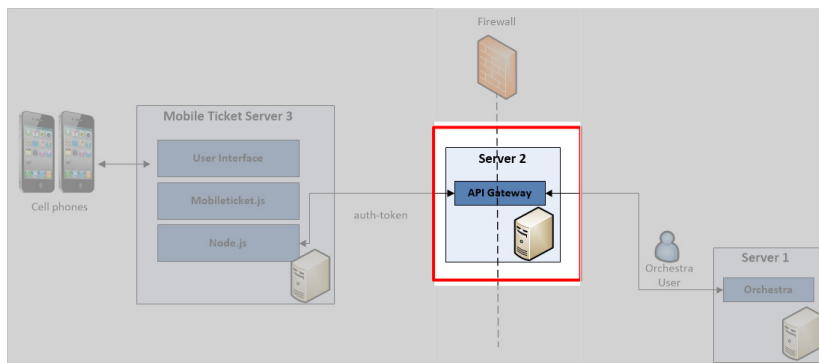
In both cases above, use Mobile Ticket URL parameter (`{{mobileFullURL}}`) in the message.

- **Appointment confirmation or Appointment reminder:** If a customer has a booked appointment, you can send a link with the appointment confirmation or the appointment reminder so that the customer can check-in. Use the parameter `appointmentPublicId`. For full check-in URL, see the table above.

4. Installation API Gateway

Install API Gateway	18
Configure API Gateway for Mobile Ticket	19

4.1 Install API Gateway



4.1.1 Before you begin

On the machine where API Gateway is installed, your firewall must be opened to **port 9090**. For more information, see the API Gateway Manual, found on Qmatic World.

4.1.2 Install API Gateway

Install API Gateway version 1.3.2.0, or later. Installation file and API Gateway manual can be found in Xtend.

4.2 Configure API Gateway for Mobile Ticket

If you are planning to use API Gateway for Mobile Ticket, follow these steps:

1. In the `application.yml` file, change the url to the Orchestra server so that it points to your Orchestra server instead.
2. Generate an API token, by running the script **api-token-generator.bat**, located in the **bin** folder of the `QP_APIGateway_win64-<_number>.zip`. For more information, see the API Gateway Manual, found on Qmatic World.
3. Enter the generated API token at the mobile user in the `application.yml` file.
4. Create an encrypted password for the mobile user, by running the script **password-encoder.bat**. For this, use the password that you entered in step 2 of the Orchestra configuration. For more information, see the API Gateway Manual, found on Qmatic World.
5. Enter the encoded password, at the mobile user, in the `application.yml` file.

5. Installation and Upgrade

Topics

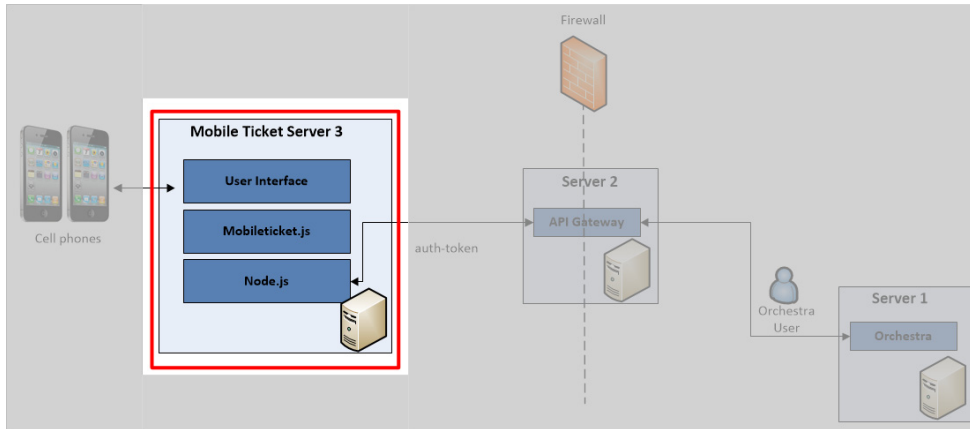
Install Mobile Ticket	22
Enable HTTPS / SSL	24
Install Mobile Ticket as a service	27
Database configuration	28
Upgrade Mobile Ticket	30

5.1 Install Mobile Ticket

5.1.1 Step 1: Install node.js

Download and install **node.js** from <http://www.nodejs.org>.

! Make sure that you select the LTS!



5.1.2 Step 2: Configure the proxy/webserver.

1. In the Mobile Ticket folder, open the file **proxy-config.json**.
2. Configure the following settings:
 - **apigw_ip_port** - API Gateway host address (IP and port).
 - **local_webserver_port** - Local webserver port.
 - **auth-token** - Mobile API User authentication token. Superadmin user should not be used.
 - **embed_iFrame** - To prevent clickjacking, the use of iFrame is blocked by default. If you need to allow iframe, set to 'true'.

! For SSL and HTTPS configuration, please see "5.2 Enable HTTPS / SSL" on page 24.

5.1.3 Step 3: Install Mobile Ticket

1. Inside the source folder, go to 'upgrade-helper' folder. (e.g: `>cd <installation_path>\mobile-ticket-1.9.2\upgrade-helper`)
2. Execute `npm install` to install all dependencies (one-time execution).

-
3. Execute below cmd to install (replace source and destination paths accordingly):

Windows

```
>gulp install --src <installation_path>\mobile-ticket-1.10.0 --dest  
<installation_path>\mobile-ticket-1.9.2
```

Linux

```
>gulp install --src <installation_path>/mobile-ticket-1.10.0 --dest  
<installation_path>/mobile-ticket-1.9.2
```



All content of the replaced folder will be replaced with the new build.

4. Open a browser and go to the ip address and port that you have configured to see if Mobile Ticket is working

Example: <http://localhost:81>

If this fails, please see “8.1 Troubleshooting guide” on page 60.



The application needs to have internet access on first start-up in order to fetch dependencies.



If you want to install Mobile Ticket as a service, please see “5.3 Install Mobile Ticket as a service” on page 27.

5.2 Enable HTTPS / SSL

5.2.1 Export certificate file and private key file from keystore

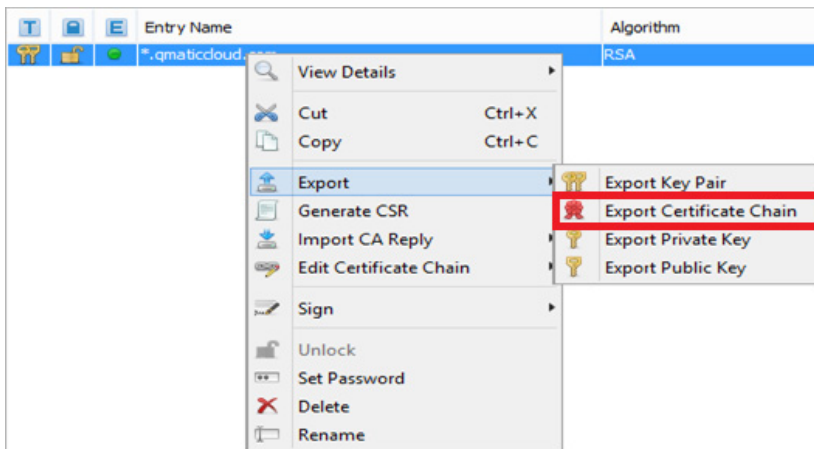
ⓘ Since Mobile Ticket uses X509 certificate format, any certificate that is created should be according to the X509 format.

ⓘ We recommend using a tool such as Keystore Explorer, <http://keystore-explorer.org/>, for managing your keystores and certificates.

Export an X509 certificate from a p12 keystore file

This certificate can then be used with Mobile Ticket.

1. Open the keystore file in Keystore Explorer.
2. Right-click and select **Export Certificate Chain**:

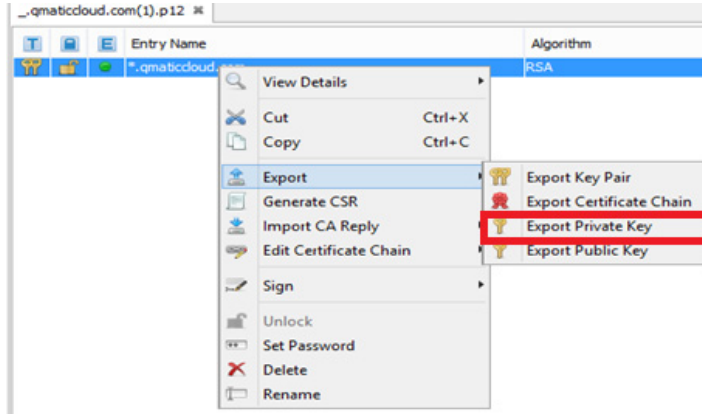


3. Select **X.509** in the popup window and keep **PEM** checked.
4. Change the file name to **server.crt** and click **Export**.
5. Copy the exported **server.crt** file to the **sslcert** folder in your Mobile Ticket installation.

Export a private key from a p12 keystore file

This private key can then be used with Mobile Ticket.

1. Open the keystore file in Keystore Explorer, right-click and select **Export Private Key**:



2. In the popup, select **OpenSSL** and click **OK**.
3. In the following popup window, deselect the **Encrypt** option and keep **PEM** checked.
4. Change the file name to **server.key** and click **Export**.
5. Finally, copy the **server.key** file to the **sslcert** folder in your Mobile Ticket installation.

5.2.2 Configure HTTPS for API Gateway

If you want to run Mobile Ticket against API Gateway in HTTPS mode, open **application.yml** file and change the **server** section so that the following parameters are uncommented:

- ssl
- key-store
- key-store-password
- key-password

For example:

```
server:
  # define server port for gateway
  port: 9090
  ssl:
    key-store: classpath:keystore.jks
    key-store-password: secret
    key-password: password
```

⚠ Make sure that you are using a keystore with a correct password.

⚠ Make sure that you have configured the **proxy-config.json** file correctly, see below section.

5.2.3 Configure HTTPS for node.js / Mobile Ticket

1. In the MobileTicket folder, open the **proxy-config.json** file.
2. The following parameters need to be configured if you want to use SSL / HTTPS:
 - **local_webserver_ssl_port** - Local webserver HTTPS port.
 - **support_ssl** - Set to true if server should support SSL. For this, you need a valid public key and a certificate in the *ssl/cert* folder. For more information, see “5.2.1 Export certificate file and private key file from keystore” on page 24.
 - **gateway_has_certificate** - Set to true if API Gateway is configured to run on HTTPS.
 - **gateway_certificate_is_valid** - Set to true if the configured SSL certificate in the API Gateway is a valid one.

⚠ This setting is only active if the `gateway_has_certificate` parameter is set to 'true'. This must be set to 'false' if the API Gateway is configured with a self-signed certificate.

- **tls_** - You only need to configure this if you are using another TLS version than the latest. Otherwise leave empty.
- **cipher_set** - Indicates cipher set to use in the server. Enter the value like this: ['ECDHE-RSA-AES128-SHA256', 'DHE-RSA-AES128-SHA256', 'AES128-GCM-SHA256']. Keep it empty if you don't want to change the cipher set.
- **hsts_expire_time** - The HSTS (HTTP Strict-Transport-Security) expire time.
- **allow_origin** - Indicates whitelist origins to use in the server. Enter the value like this: ['https://www.origin1.com', 'https://www.origin2.com', 'https://www.origin3.com']. Keep it empty if you want to allow only the same origin.

5.3 Install Mobile Ticket as a service

In production environment you may want to install Mobile Ticket as a service.

5.3.1 Install service in Windows

To install Mobile Ticket as a service for Windows, follow these steps:

1. After running `npm start` and checking that Mobile Ticket works, exit the command prompt.
2. Go to bin folder (`cd <mobileticket_installation_directory>/bin`) and run **install-mt-service.bat** file (you may need to run it as administrator). This will create a Windows service with service name "Qmatic Mobile Ticket".

To uninstall, run **uninstall-mt-service.bat**. It should then be removed from Windows service list.

5.3.2 Install service in Linux - CentOS 7

For Linux, follow these steps:

1. After running `npm start` and checking that Mobile Ticket works, exit the command prompt.
2. Install the pm2 package by executing

```
sudo npm install -g pm2
```


(<https://pm2.keymetrics.io/docs/usage/quick-start/>)

3. Execute

```
pm2 start /<mobileticket_installation_directory>/  
server.js --name="mobile-ticket"
```

4. Execute

```
pm2 startup
```

5. To install, execute

```
npm install -g pm2 && pm2 update
```

Mobile Ticket should now be installed as a service in CentOS. You can execute `pm2 list` to verify.

To uninstall the service, execute "pm2 delete "mobile-ticket".

5.4 Database configuration

Some features in Mobile Ticket requires a database and configuration related to the database. The database should be a Mongo Database and it can be any of following:

- On-premise database connection (Local Database)
- Database hosted in a cloud (e.g. AWS)
- Global cloud database service for MongoDB (e.g. Atlas)
- Docker container for MongoDB

❗ If you want to run the Mongo instance on a docker container, please see “5.4.2 Set up MongoDB on docker” on page 28.

5.4.1 Configure database settings

1. Go to `<mobileticket>/mt-service/src/config` and open **config.json**.
2. Locate **db_connection_string** and specify a valid database connection URL.

Example connection string:

```
mongodb://YourUsername:YourPassword@host:port/  
YourDatabaseName[?options]
```

3. If you are a Mobile Ticket installer / DevOps engineer who want to configure several Mobile Ticket instances to use the same database (shared database) with a discriminator for each mobile ticket instance, locate **tenant_id** and specify a unique Tenant ID (e.g. “b14f3c08”, “client-0001”).

5.4.2 Set up MongoDB on docker

This section describes how to run Mongo instance on a docker container.

1. Pull an image which gives you a mongo DB host. You can use an image called “**mongo**”
\$ docker pull mongo:latest
2. Copy the docker compose file and the DB script to a directory (required files can be found below).
3. Run the docker compose file with **\$ docker-compose up -d**

Now you should be able to see the service running. You can check it from docker desktop or **\$ docker ps**.

5.4.3 Connection strings

- Connect with root user: `mongodb://root:adminhe11@127.0.0.1:27017/mt?authSource=admin`
- Connect with db user: `mongodb://mobileticket:nsavip@127.0.0.1:27017/mt?authSource=mt`

5.4.4 docker-compose.yml

```
version: '3.8'
services:
  # ... other configs ...
  mongodb:
    image: 'mongo'
    container_name: 'mt-mongo-container'
    environment:
      - MONGO_INITDB_DATABASE=mt # database name
      - MONGO_INITDB_ROOT_USERNAME=root # set your container root username
      - MONGO_INITDB_ROOT_PASSWORD=adminhe11 # set your container root password
    volumes:
      - ./init-mongo.js:/docker-entrypoint-initdb.d/init-mongo.js:ro
      - ./mongo-volume:/data/db
    ports:
      - '27017:27017'
  # ... other configs ...
```

5.4.5 init-mongo.js

```
db.createUser(
  {
    user      : "mobileticket",
    pwd       : "nsavip",
    roles     : [
      {
        role   : "readWrite",
        db     : "mt"
      }
    ]
  }
)
```

5.5 Upgrade Mobile Ticket

This section explains how upgrade to a newer version of Mobile Ticket can be done without resetting the current configuration data.

ⓘ You can also upgrade by installing a newer version of the software. In this case, any configuration and customization performed must most likely be performed manually again.

To upgrade Mobile Ticket using the upgrade helper, follow these steps:

1. Go to the “upgrade-helper” folder inside the source folder. (e.g: `>cd <installation_path>\mobile-ticket-1.10.0\upgrade-helper`)
2. Execute `npm install` to install all dependencies (one time execution).
3. Execute below cmd to upgrade (replace source and destination paths accordingly):

Windows:- `>gulp upgrade --src <installation_path>\mobile-ticket-1.10.0 --dest <installation_path>\mobile-ticket-1.9.2`

Linux:- `>gulp upgrade --src <installation_path>/mobile-ticket-1.10.0 --dest <installation_path>/mobile-ticket-1.9.2`

All files will be replaced except below list of files/folders

- `./proxy-config.json`
- `./mt-service/src/config/config.json`
- `./src/app/config/config.json`
- `/src/app/locale/en.json`
- `/src/app/resources`

Above files and folders will be compared and if there are any new items it will be added into the respective destination.

ⓘ Upgrade is not supported if you upgrade Mobile Ticket with older branding to Mobile Ticket with new branding settings.

6. Configuration

Topics

About configuration of Mobile Ticket.....	32
Change which information is displayed	32
Change functional settings	35
Change appointment settings	39
Allow personal information.....	40
Configure open hours	42
Localize Mobile Ticket	44
Manage notification sound	46
Manage geofencing settings.....	38
Manage background time intervals	48

6.1 About configuration of Mobile Ticket

Most settings for Mobile Ticket can be set in **config.json**, located in `MobileTicket\src\app\config`.

❗ Changes to `config.json` are implemented directly, no restart is needed.

6.2 Change which information is displayed

6.2.1 Show queue position / appointment time

Above the ticket, in the queue status bar, you can select if you want to show queue position or appointment time. In `config.json`, locate and change the following parameters:

- `show_queue_position` - Show or hide queue position
- `show_appointment_time` - Show or hide appointment time for appointment visits.

❗ If estimated waiting time is enabled, the queue position is based on the queue simulation position, taking into account the work profiles of the logged in staff users. If it's not enabled, only the position in the current queue is shown.

Both queue position and appointment time cannot be shown for the same ticket. The logic for which is shown is as follows:

- If both are enabled and it's a drop-in visit -> Queue position is shown
- If both are enabled and it's an appointment -> Appointment time is shown
- If only queue position is enabled and it's a drop-in visit -> Queue position is shown
- If only queue position is enabled and it's an appointment -> Queue position is shown
- If only appointment time is enabled and it's an appointment -> Appointment time is shown
- If only appointment time is enabled and it's a drop-in visit -> Nothing is shown
- If none are enabled -> Nothing is shown

6.2.2 Show or hide number of waiting customer

To show or hide the number of waiting customers in service list, open `config.json` and locate the following parameter:

- `show_number_of_waiting_customers`

6.2.3 Show estimated waiting time

It is possible to show the estimated waiting time. The estimated waiting time is calculated for each individual customer and is based on:

- the number of open service points
- the used work profiles of the logged in users.
- the estimated service duration
- the number of people waiting

The estimated waiting time can be shown in the list of services and above the ticket number.

ⓘ The estimated waiting time is displayed as 0 until a counter opens. This can be important to keep in mind if you allow to take mobile tickets before the actual opening hours.

Please select a service at MyBranch:

Driving license with a very long... 4

Estimated waiting time: 55 min

6
You 5 4 3 2 1

Estimated waiting time: 5 min

A021
Accounts Check

Colombo

Before you begin

- You must have a Customer Journey Management premium license.
- In Business Configuration > Services, make sure all services available in Mobile Ticket have a serving time set. This is the estimated time it takes to serve this service. Make sure this time is as accurate as possible by comparing with earlier data.

- Make sure Queue Simulation application is installed and that the unit type is added to the wanted operation profiles. The extension can be found in Xtend.
- API Gateway version 1.5.4.0 must be installed.

Step 1: Enable estimated waiting time

In config.json, there are two settings to enable estimated waiting time:

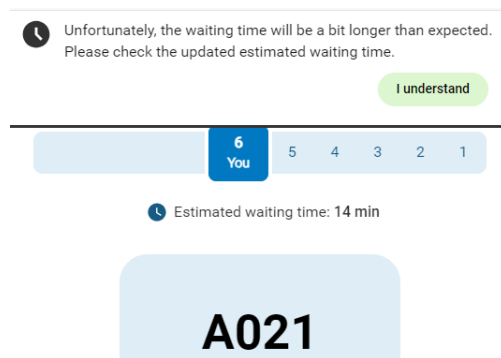
- `show_waiting_time_ticket`: If enabled, estimated waiting time is shown above the ticket card. Also, the queue position is based on the queue simulation.
- `show_waiting_time_service`: If enabled, estimated waiting time is shown in the list of services

Step 2 (optional): Change settings for estimated waiting time

In config.json, change the following settings related to estimated waiting time:

- `refresh_interval`: The time in seconds in which the estimated waiting time is updated. Default: 5 seconds.
- `information_alert`: If enabled, a popup is shown to the user if the estimated waiting time changes significantly. Default: enabled.
- `alert_time_difference_threshold`: Time in minutes which defines when the alert for changes to the estimated waiting time is shown. Only relevant if the information alert is enabled. Default: 5 minutes.

For example, if the information alert is enabled and the alert time difference threshold is set to 5, this popup is shown if the estimated waiting time is delayed by 5 minutes:



If the estimated waiting time is shortened, a similar message is shown.

6.2.4 Show accessibility statement

It's possible to allow customers to see the accessibility statement. See "9.2 End user walk-through" on page 65 to see where in the user interface it is located.

To show the accessibility statement, open `config.json` and locate the following parameter:

```
"wcag_statement": {  
  "value": "disabled",
```

6.2.5 Show custom field

To show information from a custom field in the ticket page, open `config.json` and locate the parameter below. As value, enter the name of the custom field (e.g. `custom3`):

```
"display_custom_field": {  
  "value": "custom3",
```

Leave the value empty if you don't want to display a custom field.

The customer field text will be displayed below the ticket number.

6.3 Change functional settings

6.3.1 Enable service groups

It is possible to arrange services in service groups on the page where the customer selects a service.

❗ For this feature to work, service groups must be configured in the extension General Admin. For more information, see the General Admin manual.

To enable service groups:

1. In `config.json`, locate the `service_group` parameter and set it to 'enable'.

```
"service_group": {  
  "value": {  
    "availability": {  
      "value": "enable",
```

2. Control how many service groups the user can expand at the same time with the `single_selection` parameter. If enabled, the user can only expand one service group. If disabled, the user can expand all of them.

```
"single_selection": {
```

```
"value": "enable",
```

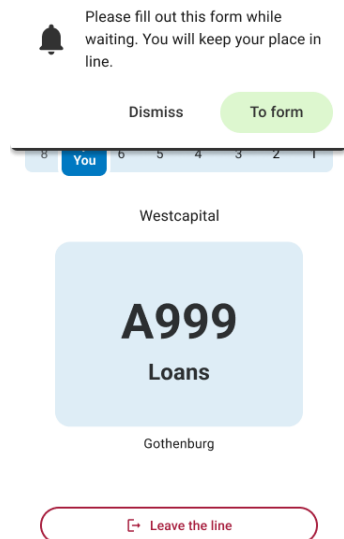
6.3.2 Link to external web page or form

It's possible to link to an external web page, such as a form that the customer needs to fill out before the appointment. A pop-up box appears in Mobile Ticket, where the customer can select a button that will take them to the web page or a form.

1. In `config.json`, locate `dynamic_url` and enter to the URL you want to link to. Use `{branchId}` and/or `{visitId}` to pass on the branch id and visit id to the URL.

```
"dynamic_url": {  
  "value": "http://www.yourdomain.com/form/  
  {{branchId}}/{{visitId}}"
```

2. To change the text in the pop-up, locate the language file (e.g. `en.json`) and edit the text.



6.3.3 Prevent taking new ticket

In `config.json`, locate `create_new_ticket` and set to 'disable' to block access to creating visits using the Mobile Ticket UI. This means the **Get new ticket** button will not be visible and access to branch and service pages (through url) is blocked.

6.3.4 Enable visit delay

In Mobile Ticket, there is a possibility to let customers get in line remotely, estimate an arrival time, and then get called on their estimated arrival time.

⚠ This feature does not work for 4.3.0.93 OG, 4.3.0.161 UD1 or 4.3.0.163 UD2.

The visit delay feature lets the customer estimate an arrival time in Mobile Ticket. Their waiting time will start counting, but they will not be called before the selected time has passed. The customer will see a countdown timer with how much time is left and can choose to extend the estimated arrival time even further.

The image displays two side-by-side screenshots of the Qmatic Mobile Ticket interface.

Left Screenshot: The header is "QMATIC". Below it, the text says "Please select a service at Berlin:". There is a vertical list of service buttons: "Accounts Check" (1 person icon), "Accounts Creation" (1 person icon, highlighted in blue), "Cash Withdrawal" (1 person icon), "Cheque Deposit" (0 person icon), and "Consultancy" (0 person icon). At the bottom, there are two dark buttons: "→ Get in line now" and "Estimate arrival time".

Right Screenshot: The header is "QMATIC". Below it, the text says "When do you want to be served?". There are two light blue buttons: "1 min" and "5 min". At the bottom, there are two dark buttons: "Confirm" and "Back".

To turn on this feature:

1. In `config.json`, locate the `delay_visit` parameter and set it to 'enable'.

```
"delay_visit": {  
  "value": {  
    "availability": {  
      "value": "enable",
```

2. Define which should be the available time slots. Enter comma-separated values in minutes.

```
"time_slot": {  
  "value": [],
```

6.3.5 Manage geofencing settings

To manage geofencing settings, open `config.json` and configure the following settings:

- `geo_fencing` - Set to 'enable', 'disable' or 'mandatory'. For geofencing to work, HTTPS needs to be enabled. If set to enabled, customers can still choose not to share location and a full branch list will be displayed. If set to mandatory, users must share location, otherwise they cannot continue.
- `branch_radius` - Define a value in meters to include branches within on the *Branch Selection* page.
- `appointment_arrive_radius` - Define a value in meters if you want to enforce geofence when appointment check-in is used.

6.3.6 Redirect to Customer Feedback

It's possible to redirect the customer to Customer Feedback when the visit is over. In `config.json`, locate `customer_feedback` and enter the Customer Feedback URL.

The URL should look like this example where the `xxxxxx` part should be replaced by the id that you got when ordering Customer Feedback.

`https://cf.qmatic.cloud/xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/index.html`

6.3.7 Show QR code

Customers can verify their visit by showing a QR code that can be scanned. The QR code holds the ticket number. You can then scan the QR code in Concierge and find the visit.

To enable showing a QR code, open `config.json` and set `show_QR` to 'enable'.

6.4 Change appointment settings

6.4.1 Enable rescheduling link

When you send out appointment confirmations by email to customers, you normally want to offer customers the possibility to:

- cancel or reschedule their appointment
- check in for their appointment.

Since this is done in different applications, you would need to provide 2 URLs. For a more seamless journey, you can link to the page for rescheduling/canceling the appointment from Mobile Ticket.

To link customers to the rescheduling page from Mobile Ticket:

1. Open `config.json` and locate the `reschedule_link` parameter.
2. Enter the URL, for example `http://localhost:8090/qmaticwebbooking/`

```
"reschedule_link": {  
    "value": "http://localhost:8090/qmaticwebbooking/",
```
3. The **Reschedule/Cancel** button will be shown below the **Arrive** button. If you leave the parameter empty, the button will not be shown.

6.4.2 Configure appointment check-in

1. In User Management, make sure there is a user called *mobile* with the same credentials as the mobile user in System Administration. The user needs to have access to all branches and be assigned to a role with the Appointment and Connector Entrypoint access modules.
2. Configure the following settings in `config.json`:
 - **appointment_late** - the time in minutes a customer is allowed to arrive late.
 - **appointment_early** - the time in minutes a customer is allowed to arrive early
 - **appointment_time_API** - Enable to get the current time at the appointment timezone (<https://timeapi.io/>)
 - **timeFormat** - time format for showing the appointment time in the check-in page.
 - **dateFormat** - date format for showing the appointment date.

6.5 Allow personal information

ⓘ Make sure the system fulfills the requirements for this functionality, see “3.1 Orchestra preparations” on page 12.

If you are going to allow customers to enter personal information, configure the following parameters in `config.json`:

- **phone_number** - Set to 'enable' to allow entering phone number, 'mandatory' to make it a mandatory field, and 'disable' to hide the field.
- **first_name** - Set to 'enable' to allow entering the customer's first name, 'mandatory' to make it a mandatory field, and 'disable' to hide the field.
- **last_name** - Set to 'enable' to allow entering the customer's last name, 'mandatory' to make it a mandatory field, and 'disable' to hide the field.
- **customer_id** - Set to 'enable' to allow entering customer ID or another customer-specific token. The data entered in customer ID will be saved in custom field 2 (can be enabled in the unit type to show in Counter).
- **privacy_policy** - Enable or disable showing privacy policy when the customer enters personal information

If you enable the privacy policy, you need to decide if you want to use an integrated privacy policy or link to an external privacy policy. If you want to link to an external privacy policy, add a link to **privacy_policy_link**. If you want to use the integrated text, leave the link setting empty and edit the text for privacy policy in the translation file (`en.json`). You can use inline css to style the text.

- **active_consent** - Select whether customer need to actively agree to the privacy policy.
- **country_code** - Specify the country code that should prepopulate the phone number field. Use either ISO alpha-2 format or International phone number format.
- **preferred_country_list** - Define country codes you want to show up in the top of the drop-down list when selecting a country code. Use ISO alpha-2 format. To add several country codes, add a comma between them.

QMATIC

Please enter your phone number so that we
can send an SMS when you are called.

Phone Number:

 +94 112 345 678

Search Country

-  Sri Lanka (ශ්‍රී ලංකාව) +94
-  Afghanistan (افغانستان) +93
-  Albania (Shqipëri) +355
-  Algeria (الجزائر) +213

[store your phone number.](#)

- **additional_data** - Enable by setting any comma-separated string with a maximum of two values (can have only one value as well). The values have to match the query parameters of one of the following URLs.

Create ticket URL: `http://<baseURL>:81/branches/[branchId]/services/[serviceId]?value1=xxx&value2=yyy`

Service group URL: `http://<baseURL>:81/branches/[branchId]/[serviceGroupId]?value1=xxx&value2=yyy`

Branch selection URL: `http://<baseURL>:81/branches/[branchId]?value1=xxx&value2=yyy`

Mobile Ticket URL: `http://<baseURL>:81?value1=xxx&value2=yyy`

Note that value1 and value2 are the query parameters and the values will be saved to custom field 3 and custom field 4 respectively. From the unit type, you can set to show these values in Counter. If there is only one query parameter value, it will be saved to custom field 3.

6.6 Configure open hours

It is possible to configure during which open hours it should be possible to take a mobile ticket.

Opening hours can be configured in two ways. Note that you cannot combine these options.

- The first option is to control the opening hours for different services by using the Service Scheduler application and unit type.
- The second option is to control the opening hours for all branches by setting open and close times in `config.json`.

6.6.1 Configure open hours per service

To configure the open hours per service, you need to install and configure the Service Scheduler application and a unit type for Mobile Ticket.

1. Install Service Scheduler, available in Xtend.
2. In User Management, make sure there is a user called *mobile* with the same credentials as the mobile user in System Administration. The user needs to have access to all branches and be assigned to a role with the Connector ServicePoint access module.
3. In `config.json`, set the parameter `branch_schedule` to 'enable'.

⚠ When this is enabled, all values from `branch_open_hours` are ignored.

⚠ Branches that have no open services will be grayed out in the branch list.

6.6.2 Configure open hours for all branches

The open hours for all branches in your system that uses Mobile Ticket are configured in the `config.json` file, in the `branch_open_hours` parameter.

⚠ In the `config.json` file, there is a distinction between the actual opening hours, for example `open_hours.week_day1`, and the time when the this should be displayed in Mobile Ticket, `display_from` and `display_to`.

Example:

The actual store might not open their doors until 08.00 but they can open up the possibility to take tickets in Mobile Ticket 10 min before the store opens. In the same way, they might want to close the possibility to take tickets 10 min before the store closes.

This would mean that the opening hours displayed to the user would be for example 08.00-18.00, whereas if someone accesses Mobile Ticket at 07:50 it

would still be possible to take a ticket, if the `display_from` parameter has been configured to 07.50.

If the time values for a day are left empty, "Closed" will be displayed for that day. This text can be changed in the language file.

If you want Mobile Ticket to appear as closed at, for example, 14:00, you need to set the close time to 13:59.

ⓘ The open hours functionality assumes that all branches as well as all devices used to access Mobile Ticket have the same time zone.

ⓘ If a user has the branch or service page open, and the branch close time is reached, the user will still be able to create visits.

6.7 Localize Mobile Ticket

It's possible to translate the Mobile Ticket UI into any language. It is supported to offer several languages in the same installation. Both left-to-right and right-to-left languages are supported.

Which language Mobile Ticket is displayed in when it opens is determined in the following way:

- If the user opens Mobile Ticket from a link to check in for an appointment which has a language parameter, Mobile Ticket is displayed in that language.
- If Mobile Ticket opens without any language parameter defined, and Mobile Ticket has been used recently in the same browser, the user interface is displayed in that language. If it has not been used recently, Mobile Ticket is displayed in the default browser language.
- The user can also select another language from the language picker. This requires that the language picker is enabled.

❗ If the defined language does not have a matching language file in the locale folder, Mobile Ticket is displayed in the default language (English).

The displayed language has effect on if the distance to the branches is shown in metrics or imperial. This requires that the unit of distance setting is set to 'auto'.

The language used in Mobile Ticket when the visit is created is saved on the visit and is then passed on through the rest of the journey. It can be used for example for segmentation into queues and for Counter staff to know which language a customer speaks.

❗ The en.json file must never be removed from the system.

6.7.1 Translate Mobile Ticket UI

Translating the static texts in Mobile Ticket is done by translating the texts in MobileTicket\src\app\locale\en.json file.

1. Copy the en.json file to the same folder.
2. Rename the copied file to your language code.
3. Translate all the phrases in the file.

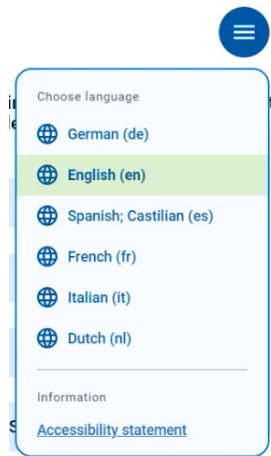
❗ Make sure that your translated file contains correct JSON format. Use a json validator to check the translated json files.

❗ If a phrase for some reason is missing in your translated json file, the fall-back language is English (en.json).

❗ If you are using the Chrome browser, you need to restart your computer after changing the locale for the settings to be applied properly.

6.7.2 Enable language picker

By default, the language displayed in Mobile Ticket is defined by the user's browser language. You can however allow users to select a language from a language picker. The languages shown in the language picker depend on the languages available in the locale folder.



- To allow users to select a language, open `config.json` and set `language_picker` to 'enable'.

6.7.3 Translate service and branch names

❗ To use this feature, General Admin is needed. See “3.1 Orchestra preparations” on page 12.

1. In `config.json`, set `service_translation` to 'enable'.
2. In the file `application.yml` for the API Gateway installation, add the following lines under `routes`:

```
translations:
  path: /MobileTicket/translations/*
  url: ${orchestra.central.url}/calendar-backend/api/v1
```

3. In General Admin, make sure all service and branch names are translated.

6.7.4 Configure unit of distance

You can configure the unit of distance shown in the branch list page.

1. Open `config.json` and locate the `system_of_units` parameter.
2. Set to 'metrics', 'imperial' or 'auto'. Auto means the distance unit will be based on the browser locale.

6.8 Manage notification sound

When a customer is called, a notification sound is played. This sound can be changed:

1. In the `MobileTicket\src\app\resources` folder, add the sound file you wish to use (default file is `notification.mp3`).
2. In `config.json`, locate the `notification_sound` and enter the name of the new sound file.

6.9 Enable Google Analytics

If you use Google Analytics, you can use this to track Mobile Ticket. If Google Analytics is enabled you may need to ask for cookies consent. If you need to ask for consent depends on which region you are in.

1. In `config.json`, locate the parameter `ga_track_id` and enter your Google Measurement Id.
2. To enable active cookies consent, set `cookie_consent` to 'enable'. If enabled, a pop-up will be displayed until the user accepts or declines.
3. To change the cookies information page that the cookie consent message links to, open and edit `locale/cookie-consent-files>cookie-<LANG>.html`.

When showing the cookies information page, the browser will pick language file depending on language set in the browser.

6.10 Configure security settings

It's possible to enable features for extra security to prevent systematical misuse and hacker attacks.

6.10.1 Prevent opening in another device

You may want to limit the possibility to open Mobile Ticket in another device than the one it was created in.

- To prevent a mobile ticket from being opened in another device, set `block_other_browsers` in `config.json` to 'enable'.

6.10.2 Use one-time passwords

You may want to send a one-time password (OTP) to all users who want to create a mobile ticket to confirm that they are a unique user.

The flow is like this: The user is asked to enter phone number > A PIN number is sent to the user > The user enters PIN.

Before you begin

- This feature uses a database, so you need to set up the database configuration. See "5.4 Database configuration" on page 28.
- You need to have a mobile user in Orchestra with same credentials as in System Parameters and access to all branches. The user needs to be assigned to a role with the Notification Service access module.

Enable one-time password functionality

1. Configure API Gateway by adding `sms_api` mapping to `application.yml`

```
sms_api:
  path: /rest/notification/**
  url: ${orchestra.central.url}/notification/
```
2. Go to `src/app/config/config.json` and set `otp_service` to 'enable'.
3. Restart the server by running `npm start`.

6.10.3 Prevent replay attacks

Before you begin

- This feature uses a database, so you need to set up the database configuration. See "5.4 Database configuration" on page 28

Prevent replay attacks

If you want to prevent the possibility to create multiple tickets using replay requests, enable `create_ticket_token` in `config.json`. Then restart the server by running `npm start`.

6.11 Manage polling and retry settings

You can change the polling intervals and the number of retry attempts in `config.json`.

6.11.1 Service fetch interval

`service_fetch_interval` - Set the interval at which service information is refreshed. Default value is 15 seconds. This setting is only used if the setting 'Show number of waiting customers' is enabled.

6.11.2 Screen timeout for service page

`service_screen_timeout` - Set the screen timeout duration in minutes. This determines how long the app will stay in the services screen, without creating a ticket. Default value is 10 minutes.

6.11.3 Queue poll settings

`queue_poll_retry` - Set a value for how many times the queue poll request should be retried. Default value: 2.

`queue_retry_timeout` - Set a timeout value for queue poll request retry if communication to Orchestra/Qmatic Experience Cloud fails. Default value: 4.

6.11.4 Polling intervals of ticket status requests

With the following settings, you can define how often queue position should be checked. You can have different intervals depending on if queue position is above 4 or below 5.

```
"ticket_status_poll": {
  "value": {
    "over_five": {
      "value": "5",
      "description": "Interval if the ticket is place no
5 or after no 5" // position in the queue
    },
    "under_five": {
      "value": "1",
```

```
        "description": "Interval if the ticket is place  
before no 5" // position in the queue  
    }
```

6.11.5 Retry of branch and service requests

The following parameters can be adjusted to reduce the retry-load in case of failure while branch and service lists are loading. Configure `set_retry_limit` to 'enable' and then set an API request retry limit and an API request retry interval.

```
"api_retry": {  
    "value": {  
        "set_retry_limit": {  
            "value": "enable",  
            "description": "enable or disable retry limit for  
an api request"  
        },  
        "retry_limit": {  
            "value": "10",  
            "description": "Maximum number of retries for an  
api request"  
        },  
        "retry_interval": {  
            "value": "5",  
            "description": "Interval (in seconds) between the  
two api requests when one request failed and retried  
again."  
        }  
    },  
    "description": "Configurations related to retrying api  
requests"  
},
```


7. Branding

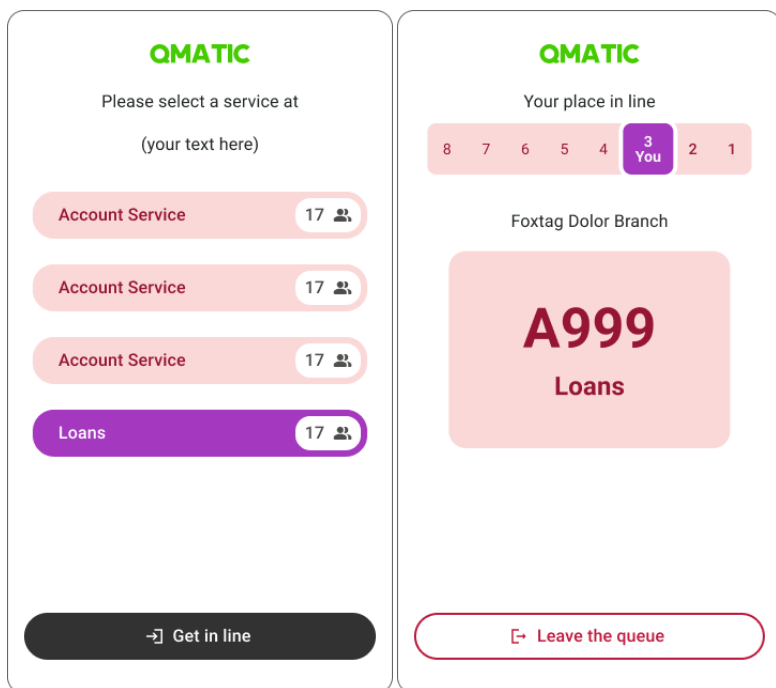
Introduction	2
Accessibility guidelines	4
Customize logo and colors	5

7.1 Introduction

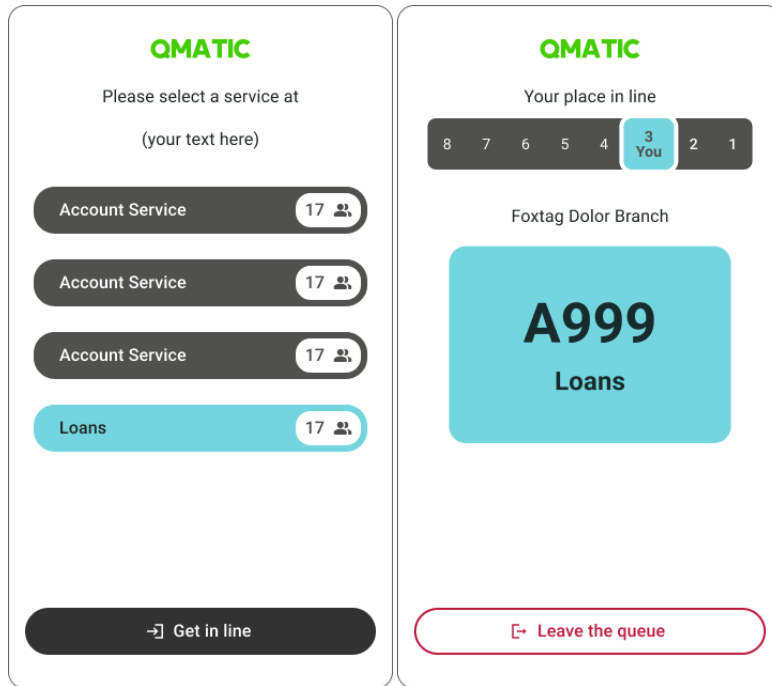
Mobile Ticket can be adapted to match your brand guidelines by including your own logo and choosing colors that match your color scheme.

7.1.1 Branding examples

This first example has a light pink as primary color and dark red as primary text color. The secondary color is purple and the secondary text is white.



The second example has a dark gray primary color with a white primary text color, and a turquoise secondary color with a dark secondary text color.



⚠ If you need help choosing a color scheme, there are lots of tools online that can generate color schemes.

7.2 Accessibility guidelines

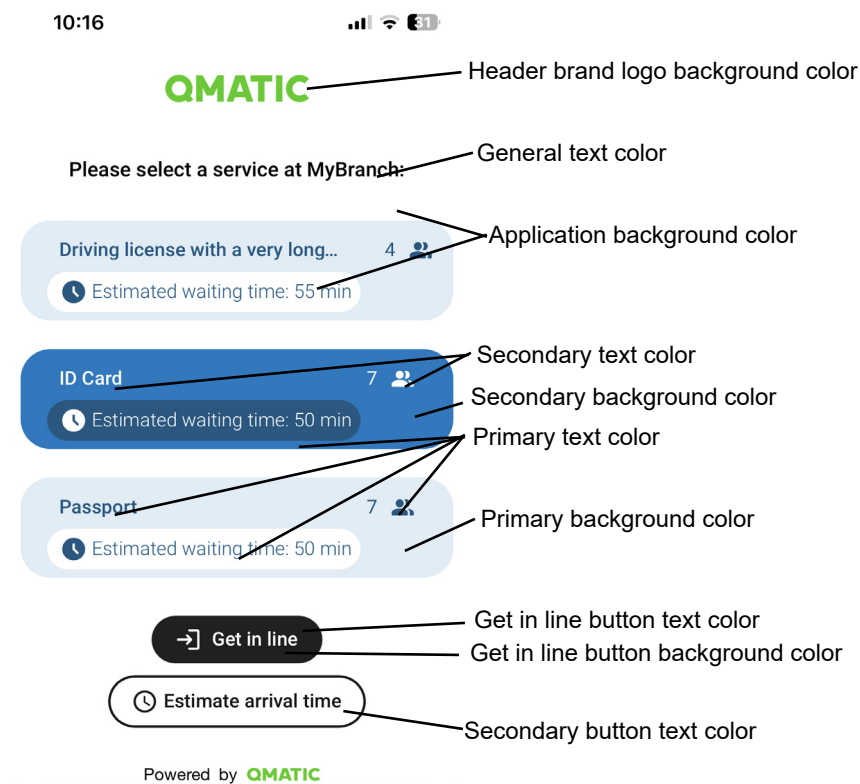
We strongly recommend that the color combinations you choose pass the Level AA contrast requirements defined by WCAG. There are free tools online that can be used to evaluate the color contrast. To meet the accessibility requirements for level AA, the contrast ratio between the text color and the background color (the button, the ticket, etc) needs to be more than 4.5.

A recommendation is that you choose white or black for the text color. You can choose dark or light shades of other colors, but make sure you meet the requirements.

If you choose colors that don't pass the contrast requirements, the provided accessibility statement is not valid anymore.

7.3 Color reference information

Selecting colors is important for brand recognition, but also for usability and accessibility. Make sure you select and test colors carefully.



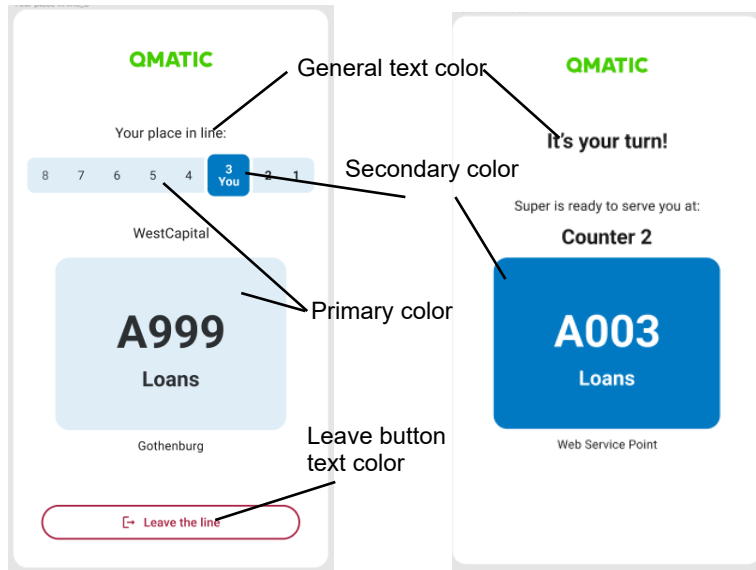


Table 1: CSS file properties

Element name	Property name in css file	Description
Header brand logo background color	brand-logo-background	Sets the background color of the logo in the header.
Application background color	app-bg-primary-color	Background color in the application. Ensure good contrast to the other user interface components when choosing the background color.

Table 1: CSS file properties

Element name	Property name in css file	Description
Primary background color	primary-bg-color	Background color for services and branches before they are selected, delay time and appointment arrival, and for the ticket in the waiting page, etc. By default, the primary color is light blue.
Primary text color	primary-text-color	Text color for inactive (unselected) services, branches, delay time, appointment arrival, ticket layout in waiting page, etc. If you have chosen a dark color as primary background color, we recommend a light primary text color and vice versa.
Secondary background color	secondary-bg-color	Background color for active (selected) services, branches and delay time, and for the ticket in the calling page, etc. By default, the secondary color is blue.
Secondary text color	secondary-text-color	Sets the text color for active (selected) services, branches and delay time, and for the ticket in the calling page, etc.
Service group text color	service-group-text-color	Text color of the displayed service groups.

Table 1: CSS file properties

Element name	Property name in css file	Description
Service group line color	service-group-line-color	Color of the line separating the displayed service groups.
General text color	general-text-color	Color of the text used throughout the application that are not specific to any UI component.
Footer text color	footer-text-color	Color of the text near the footer logo.
Popup button text color	pop-up-button-text-color	Text color of the consent button in privacy and cookie popups.
Get in line button background color	get-in-line-btn-bg-color	Background color for the “get in line” button at the bottom of the display.
Get in line button text color	get-in-line-btn-bg-text-color	Text color for the “get in line” button at the bottom of the display.
Leave button text color	leave-button-text-color	Text color of the “Leave the line” button.
Secondary button text color	secondary-btn-text-color	Text color of secondary button such as “Estimate arrival time”, “Send information” and “Skip this”.

7.4 Customize colors

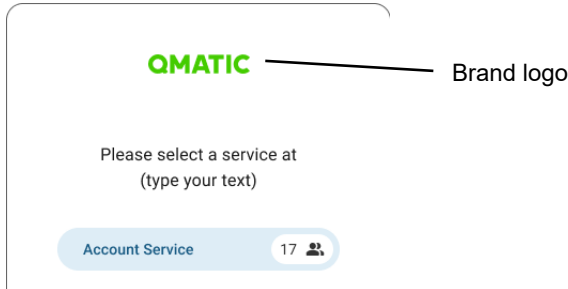
Colors and logo are configured in the file `theme-styles.css`

1. In `mobile-ticket-<version>/src/app/theme`, open `theme-styles.css`
2. Change the css properties according to your color scheme.
3. Save the file. Changes are applied immediately.

```
theme-styles.css
1 :root {
2   /* Header brand logo image */
3   --brand-logo:url('../app/resources/qmLogo.png');
4   /* Header brand logo background color */
5   --brand-logo-background:transparent;
6   /* Application background color */
7   --app-bg-primary-color: #FFFFFF;
8
9   /* Background color for Services,Branches,delay time, appointment arrival, ticket layout in waiting page, etc */
10  --primary-bg-color:#DEEDF6;
11  /* Text colour for Services,Branches,delay time, appointment arrival, ticket layout in waiting page, etc */
12  --primary-text-color:#000000;
13  /* Background color for Services,Branches,delay time when selected and ticket layout in calling page, etc */
14  --secondary-bg-color:#0079C2;
15  /* Text color for Services,Branches,delay time when selected and ticket layout in calling page, etc */
16  --secondary-text-color:#FFFFFF;
17  /* Service Groups text color */
18  --service-group-text-color:#175A83;
19  /* Service group separator line color */
20  --service-group-line-color:rgba(0,0,0,0.2);
21
22
23  /* Text color used throughout the application in common areas which are not specific to any UI component. */
24  --general-text-color: #000000;
25
26  /* Footer logo */
27  --footer-logo:url('../app/resources/brand_logo.png');
28  /* Footer text */
29  --footer-text-color: #000000;
30
31  /* Privacy & Cookie concent text color*/
32  --pop-up-button-text-color:#0079C2;
33
34  /* Bottom button/get in line button background color */
35  --get-in-line-btn-bg-color:#20201F;
36  /* Bottom button/get in line text color*/
37  --get-in-line-btn-bg-text-color:#FFFFFF;
38  /* Bottom button leave the line color*/
39  --leave-button-text-color:#A71D3D;
40  /* Secondary button text color which is blue by default */
41  --secondary-btn-text-color: #0079C2;
42 }
```

7.5 Customize logo

You can place your organization's logo at the top, clearly communicating your brand. Our recommendation is that you do not have padding (white area) around your logo, since that will make the logo smaller.



The size of the logo should be 1000 px wide or 500px high, whichever of these two values that is the highest. The logo must not exceed the max height nor the max width.

Mobile Ticket is adapted to high resolution smartphone displays. If Mobile Ticket is used on other devices, make sure that the brand logo does not appear with pixels on that device and that the logo is readable. This is to ensure a reliable and professional appearance.

7.5.1 Set a top logo

1. Place the image file with the logo in `mobile-ticket-<version>/src/app/resources`
2. In `src/app/theme`, open **theme-styles.css** and enter the logo filename in `brand-logo`.

7.5.2 Customize the footer

By default, Mobile Ticket shows a footer with the text "Powered by Qmatic" where "Qmatic" is a png file with the Qmatic logo.

The footer can be customized with the following settings in `config.json`:

- **logo** - Enable or disable the brand logo in the footer. This requires that you have added and configured a brand logo.
- **custom_text** - If you don't want to use an image as logo, you can just enter the company name here as plain text. The text is inserted right after "Powered by" (which can be edited from the language file).

If you add both a custom text and enable the logo, the logo setting overrides the custom text.

8. Troubleshooting

Troubleshooting guide	2
Frequently Asked Questions	4

8.1 Troubleshooting guide

If, after following the configuration procedures, your Mobile Ticket setup still does not work as expected, we suggest that you check your configuration by following the steps below. Use a tool such as Advanced REST client or Postman to perform the REST calls.

ⓘ For more information about REST calls that can be used for the Mobile Tickets scenario, through the API Gateway, please see the API Gateway SDK documentation.

1. Perform the REST call below, to verify that you have a successful connection to Orchestra central (using port 8080)

ⓘ You need to replace “serviceld” and “branchld” with valid Id numbers from your configuration. Also, you need to use *Basic Authorization* and the *Username* and *Password* for the *mobile* user (set in *Parameters* in the *System Administration* application, in Orchestra).

/qsystem/mobile/rest/v2/services/serviceld/branches/branchld/ticket/issue/

If the REST call successfully issues a ticket, you have verified that the connection works.

If it fails, however, check that Orchestra is configured correctly.

2. To verify that your connection through the API Gateway is correctly configured, perform the following trial REST call through API Gateway.

ⓘ Make sure that you have set the correct authorization, using your *auth-token* and that you replace *[serviceld]* and *[branchld]* with valid Id numbers from your configuration.

/MobileTicket/services/[serviceld]/branches/[branchld]/ticket/issue

If the REST call is successful and a ticket is issued, you have verified that the connection works.

If it fails, however, make sure that you have configured the API Gateway correctly. See “4.2 Configure API Gateway for Mobile Ticket” on page 19.

-
3. Start the *node.js* server, from a command prompt, with the following command:

`npm start`

4. Check that the web page and UI works as expected, by entering *https://<your ip-address>:4443* in the browser address field.

8.2 Frequently Asked Questions

The table below contains some answers to some frequently asked questions:

Question	Answer
Why are my branches not available, in the branch selection list?	In Orchestra, make sure that the check box <i>Mobile enabled</i> is checked for all the Branches that you want to be listed.
Why are my services not available, in the service selection list?	In Orchestra, make sure that the check box <i>Mobile enabled</i> is checked for all the Services that you want to be listed.
Why can I not create a mobile ticket?	In Orchestra, make sure that the <i>Visi-tApp</i> unit type is installed. Make sure that API Gateway is configured correctly.
Why is my connection to Orchestra/ API Gateway not working properly?	Make sure that you have opened up port 9090, in your firewall, see “4.1.1 Before you begin” on page 18. Make sure that API Gateway is configured correctly.

9. Walk-through

Use cases	6
End user walk-through	7

9.1 Use cases

The following are examples of typical use cases:

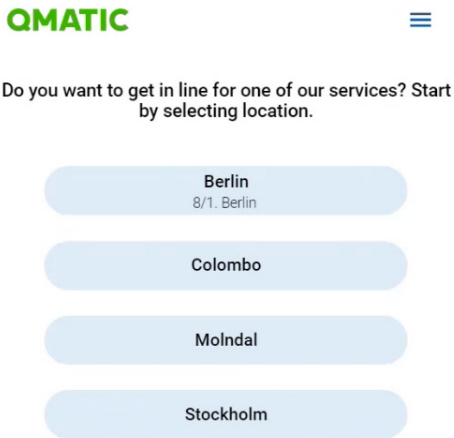
- The end-customer goes to the branch and takes a paper ticket with a QR code on it. Information on the ticket and/or in the branch says something like “If you want to use your mobile phone to stand in line, please scan the QR code!”. The end-customer scans the QR code and can then follow the progress of the line on his/her phone.
- The end-customer goes to the branch and sees information, something like “Go to this url/Scan this QR code to take a ticket online!” The end-customer then enters the url/scans the QR code, takes a ticket and can then follow the progress of the line on the phone.
- The end customer goes to the branch, and either enters his/her phone number in a kiosk when selecting service, or gives the phone number to a receptionist/floorwalker. When the visit is created, this triggers a notification with a link to that particular visit in Mobile Ticket, so that the customer can follow the progress online.
- The end customer books an appointment from home and enters contact information. An hour before the appointment he/she gets an appointment reminder with a link to check in for the appointment and follow the progress of the ticket online.

ⓘ For information about how to configure the use of QR codes and SMS notifications, please refer to “3.2 URLs for notifications and QR codes” on page 14.

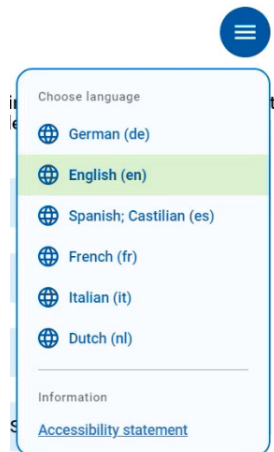
9.2 End user walk-through

In the following sections, a common flow for Mobile Ticket is described. It starts when the user has opened Mobile Ticket (via URL or QR code) with no services or branches preselected.

9.2.1 Selecting a language and a location



In the location selection page, as well as in all parts of Mobile Ticket, the user can select a language by clicking on the **More** icon .

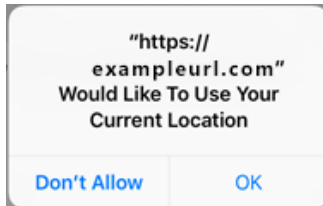


From here, the user can also select to view the accessibility statement.

The user can then select a location, represented by the branches in the system. Only branches within the defined range are listed. Branches with a defined position are sorted on distance. Branches without a defined position are sorted alphabetically.

If there are no branches within range, you are asked to try again. If there is only one available branch, within range, it will be selected and you will move on to service selection, automatically.

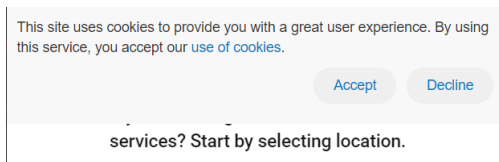
The position is defined by the browser. If using a smart phone, you will get a question that looks something like this:



ⓘ For location detection to be supported, you need to run in HTTPS.

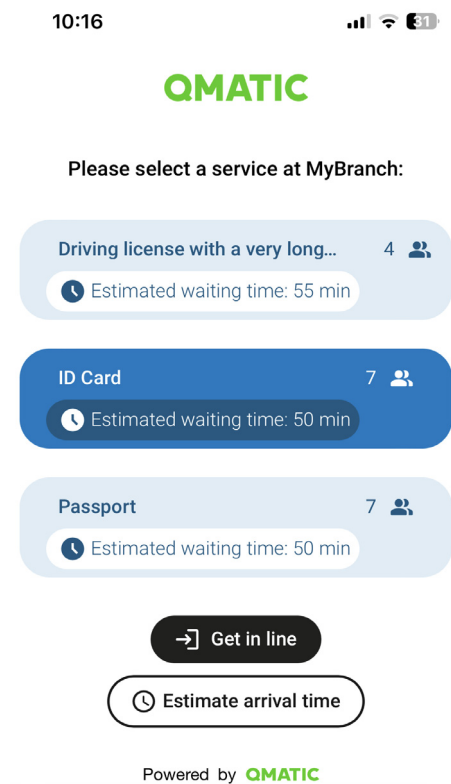
If positioning services are turned off and your position therefore is unknown, all available branches will be listed, in alphabetical order.

9.2.2 Cookies



If active cookie consent is activated, this popup will be visible until the user accepts or declines.

9.2.3 Selecting a service



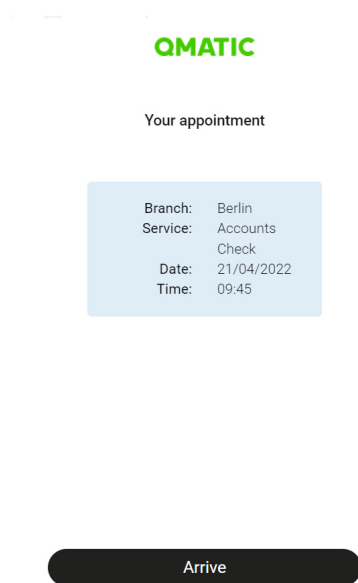
Next, the user selects the wanted service and selects Get in line. Services are sorted alphabetically and can be sorted into service groups. For each service, the estimated waiting time (if enabled) and the number of people waiting is displayed.

If only one service is available, it is automatically selected, but the user still needs to select Get in line to generate a ticket.

The user may then be prompted to enter their phone number in order to receive a PIN number, if the one-time password functionality is enabled.

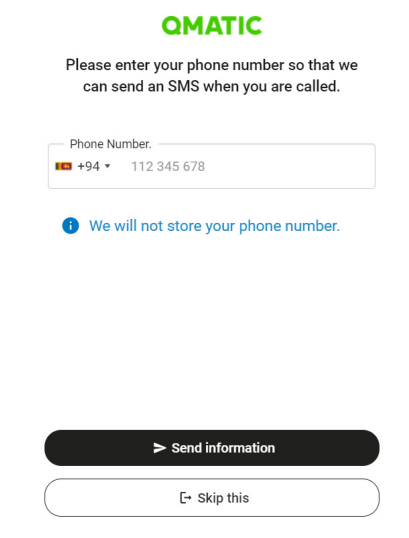
9.2.4 Alternative entry - appointment check in

An alternative flow to enter the queue is if the user uses a link to check in for an appointment. When the user has checked in, the normal visit flow will be followed.



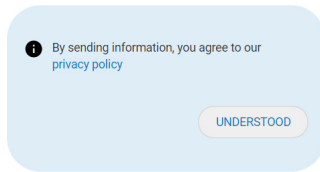
9.2.5 Add phone number

If the functionality to add phone number is enabled, that is displayed before the ticket is shown.



Before the phone number is added, the user has to agree to the privacy policy (if that is enabled)

QMATIC



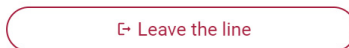
9.2.6 Waiting to be called

When the ticket has been generated, the ticket number is displayed.

QMATIC



Berlin



While standing in line, the user can follow the queue progress and place in line as her or she gets closer to being called. If it is an appointment visit, the appointment time is shown instead of the queue position. There is also an option to leave the line, which will direct the user to the Thank you page. The same will happen when using the back button in the browser.

If the visit is transferred, the information about the new service name is displayed. If the visit is recycled, a message saying "You will be called shortly" is displayed.

ⓘ Popups and alert/confirm dialogs use the default browser behavior. This means that, for example, if the user clicks **Yes** when asked if he/she wants to leave the line, this value will be cached and the popup will not be displayed again, until the browser cache is cleared. It is also possible to create customized popups, if you want to avoid the browser behavior.

9.2.7 Called to a counter

Once you are called, the ticket number will be flashing, a sound will be played (only for desktop browsers) and the user will get information about which service point and staff member to go to.

QMATIC

It's your turn!

Super is ready to serve you
at:

WebCounterMulti
Staff



ⓘ Note that the flashing ticket number and sound only will be played the first time the ticket is called. If Recall is used, no sound etc is played.

It is also possible to configure the system so that the user is notified by SMS when he or she is called.

If remote service is used (via a communication platform, e.g. Teams or Zoom), a button to open the virtual meeting is visible in the ticket page.

9.2.8 When the visit is over

When the user has been served, and the visit is ended, one of the following things will happen, depending on the configuration.

- A Thank You page will be displayed. From this page, it is also possible to take a new ticket, by selecting *Get new ticket*, if that setting is enabled. This is however not possible if the ticket was opened via a URL.

QMATIC

Thank you for visiting
us!

Welcome back!

Galle

Your ticket has been
removed

Get new ticket

-
- You are redirected to a Customer Feedback survey page.

9.2.9 Landscape Mode

For demo purposes, we recommend that you use Mobile Ticket in portrait mode. However, it is of course also possible to use Mobile Ticket in landscape mode.

Q-MATIC

Q-MATIC AB,
Box 198,
SE-431 23 Mölndal,
Sweden.

VAT No SE556212749701

Phone: +46 31 756 46 00

Email: info@qmatic.com

Web: www.qmatic.com