Lesson 2:

Switching between MapTrip and App

Introduction

This lesson is a simple one. It shows how to bring MapTrip to front or your own app.

During this lesson you will

- bring MapTrip to front
- bring your own app to front
- extend the MTI Callbacks usage

Prerequisites

You should finish the first lesson successfull to understand the basic mechanisms of MTI.

Nevertheless you can examine this lesson without considering lesson 1 because the basics like initializing the API and ensuring MapTrip is running are already implemented in lesson 1.

Like before the project has to be cloned to your computer. For testing the implementation, MapTrip should be installed at your device or device emulation.

Switching between your App and MapTrip

The MTI interface makes it easy to bring your own app or MapTrip to front. In some cases of MapTrip integration into other applications this feature is required, for example if only one of both should be in foreground.

To see how this works we only have to implement some methods and extend the MtiListener which we created before in lesson 1. To keep it simple the MtiListener is already part of the project.

Start with Project

When you begin learning with this tutorial at first (and only for one time) you should clone the tutorial project.

Cloning the MTI Tutorial

- SSH: git@github.com:MapTrip-Navigation/MTI-Tutorial.git
- HTTPS: https://github.com/MapTrip-Navigation/MTI-Tutorial.git
- GitHub CLI: gh repo clone MapTrip-Navigation/MTI-Tutorial

Open lesson 2 with Android Studio

In Android Studio Click at File menu and select the module ***lesson2_showapps***.

HowTo: Bring MapTrip to front

Extend the app with the next button

Add the next button in file file ui/home/HomeFragment.java:

```
/*
 * Within this method we will extend the button list for available functions lession by
lession
 */
private ArrayList getLessions () {
    int lessonIndex = 0;
    // Placeholder for later implemented lessons
    lessonArrayList.add(new Lesson1_Initialize(lessonIndex++,"Start MapTrip And Init
MTI", this));
    lessonArrayList.add(new Lesson2_ShowApps(lessonIndex++, "Show MapTrip 10 seconds",
this));
```

```
    return lessonArrayList;
}
```

Next bring MapTrip to front.

For this we use the API call Api.showServer().

Open the java file **Lesson2_ShowApps.java**.

Extend the overwritten method doSomething() with this lines:

```
@Override
public void doSomething() {
    // At first check if MTI is ready to use
    if (!statusInitialized) {
        return;
    }

    setMapTripToFront();
}
```

Then implement the method setMapTripToFront:

```
/*
 * show MapTrip
 */
private void setMapTripToFront() {
    Api.showServer();

}
```

After this method is called, MapTrip should come to foreground.
MTI tells us about the success of this action by raising the callback **showServerResult()**.
This callback we can use to set our own app back to front.

HowTo: Bring your own app back to front

Implement the MTI Callback showServerResult() in **Lesson2_ShowApps.java**:

```
@Override
public void showServerResult(int requestId, ApiError apiError) {
    // MapTrip is set to front and MTI raises the callback
    // From now we wait 10 seconds and switch back to our own app

    try {
        Thread.sleep(10000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // Time expired switch back
```

```
    setTutorialAppToFront();
}
```

As soon as MapTrip was set to front, the callback is raised.
To make this obvious, we wait some seconds (10.000 millis).
When the waiting period expired, we call the method setTutorialAppToFront():

```
/*
 * Bring this app back to front
 */
private void setTutorialAppToFront() {
    String className = fragment.getActivity().getClass().getCanonicalName();
    String packageName = fragment.getActivity().getPackageName();

    // MTI call
    Api.showApp(packageName, className);
}
```

For this case study it is not necessary to overwrite the matching callback. But for completion this lesson we should:

```
@Override
public void showAppResult(int requestId, ApiError apiError) {
    // here you could react to incoming callback that this app was brought to front
}
```

Testing the program at first you have to click at the button START MAPTRIP AND INIT MTI to ensure that MapTrip is running and the MTI lib is initialized.

With click at SHOW MAPTRIP 10 SECONDS MapTrip should be shown for 10 seconds and after that again your own app.

Abstract

At the end of this lesson you have created an app which starts MapTrip and initializes the MTI API. Also this app can switch between MapTrip and the tutorial app.

The code fragments mentioned in this lesson are also stored in the folder lesson2_showapps/content. This are:

- Lesson2_ShowApps.java (bring apps to front)
- HomeFragment.java (extended with a button for this lesson)
- MtiListener.java (the callback class for MTI messages)

The complete github project is hosted here: https://github.com/MapTrip-Navigation/MTI-Tutorial