

Java SIM Applet Development

Rapidly begin to Develop, Test & Debug your own VAS Applications



Creating new services has until now required long development cycles and proprietary implementation. Using Java Card based SIM cards and Gemalto development & testing tools, you can now work in-house, and transform ideas into **revenue-generating applications** in a matter of days.

This training course will allow you to rapidly get to know how to use the **Toolkit API & Developer Suite tools**, and begin developing and testing your own VAS applications.



At the end of the training you will

- > Be able to design and write your own 'interoperable' STK applet.
- > Be able to test and debug your STK applet
- > Have an overview on how avoid common traps when designing EEPROM embedded Java card applets.
- > Get some tips on how to optimize your code to gain in time and speed.

Who should attend

- > Development staff

Pre-requisites:

- > A basic knowledge of object-oriented programming is mandatory
- > It is strongly recommended that you follow the "Java SIM Card Administration" course before attending this course.

This course is held in English

Key topics

- | | |
|---|---|
| <ul style="list-style-type: none"> > CLASS CONVERSION > JAR AND CAP FILES > 43.019 API > HANDLERS | <ul style="list-style-type: none"> > BER-TLV format of Proactive commands > Examples of STK applet coding > Applet access to GSM files > Best Practices |
|---|---|

Course Schedule



Day 1	Practice
<ul style="list-style-type: none"> > Reminder on Java card architecture (JCRE, VM) > Basic Description of Java card API > Detailed Description of 43.019 API > Practical Exercises - Import existing project 	<p>Add a new STK menu to an applet implementing a proactive command, compile/convert/load/install/test</p>

Day 2	Practice
<ul style="list-style-type: none"> > Create new project & debug using card simulator > Practical Exercises > Overview on Optimisation of code in size & speed > Practical Exercises > Advanced debugging techniques 	<p>Add a new service in an applet using the event SMS_PP and other proactive commands.</p> <p>Configure an OTA communication chain to test and debug your applet.</p>

