

SERENA Setup and Installation Guide (V1, Edited 2013/04/17)

Thank you for your interest in SERENA, the Self-Reporting and Experience Sampling Assistant. This document provides a brief overview of what you can do with SERENA and guides you through the installation and preparation process if you want to setup a study using SERENA.

Acknowledgments

You can use SERENA for your research free of charge. However, we ask you to cite the following paper in your work:

A. Möller, M. Kranz, B. Schmid, L. Roalter, S. Diewald
Investigating Self-Reporting Behavior In Long-Term Studies
In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2013)*, Paris, France, April-May 2013.

If you use BibTex, you can use the following BibTex entry:

```
@inproceedings{chi2013selfreport,  
  author = {Andreas M\"oller and Matthias Kranz and Barbara Schmid and Luis Roalter and  
Stefan Diewald},  
  title = {Investigating Self-Reporting Behavior In Long-Term Studies},  
  booktitle = {Proceedings of the 2013 ACM annual conference on Human Factors in Computing  
Systems},  
  series = {CHI '13},  
  year = {2013},  
  isbn = {978-1-4503-1899-0},  
  location = {Paris, France},  
  numpages = {10},  
  publisher = {ACM},  
  address = {New York, NY, USA},  
}
```

This software is provided under the GNU General Public License (GPL) 3.0. For the full license, see <http://www.gnu.org/licenses/gpl.html> or the “license.txt” file included with the ZIP file. The software is provided as is; we do not take over any responsibility for its correct operation and cannot provide any support. Please make sure if everything works as anticipated when you use it for your research project and conduct some test runs with your configuration you are going to use.

Contact us under **andreas.moeller@tum.de**

Find more information about SERENA and our research group at

<https://vmi.lmt.ei.tum.de/serena> and <https://vmi.ei.tum.de>.

Table of Contents

- Functionality of SERENA
- Components
- Server Installation
- Conducting a Study with SERENA
 - Defining Parameters
 - Creating Questionnaires
 - Building the Application
 - Collecting the Data
 - Evaluating the Data

Functionality of SERENA

SERENA helps to collect self-reported data in long-term user studies with the smartphone. Participants enter self-reported information in questionnaires that are displayed on the phone. In addition, application usage can be monitored automatically. Logs can be used to obtain ground truth, or you can log quantitative data and additionally ask for subjective feedback through questionnaires.

All collected (log and questionnaire) information is stored on the phone and automatically transferred to the backend server. The server also enables easy analysis and export of the data.

Components

SERENA consists of two main parts: a backend server and a client application, which is installed on the participants' smartphones.

The backend server helps creating the questionnaires and receives the log data, and also visualizes the results. The questionnaire application serves for displaying questionnaires directly on the phone under definable conditions (e.g. interval- or event-based), and additionally can monitor smartphone usage in the background.

The ZIP file contains two folders “backend” and “questionnaire_app”. Additionally there are the GNU license text, the questionnaire schema, and this document as PDF.

Server Installation

The backend is written in Python and requires Python Pyramid to be installed on the server. When Pyramid is installed, just copy the “backend” folder to the desired location at the server computer. In a terminal window, change to the directory containing the “setup.py” file and execute the following commands:

```
$venv/bin/python setup.py develop
```

```
$venv/bin/populate_qserve development.ini
```

```
$venv/bin/pserve development.ini
```

It is also possible to run SERENA completely without a server, just using the Android application. However, in that case you would be required to manually create the questionnaire definitions and to collect the log files from participants' devices.

It is recommended to make use of the full server/client setup, as you benefit from assisted creation of questionnaires and automatic upload of log data.

Conducting a Study with SERENA

In the following, the necessary steps to use SERENA in your study are described. It is assumed that you already have installed the backend server as described previously in the Installation section.

The three main steps you will need to execute are: the creation of your questionnaires that shall be included in the app; the definition of necessary parameters, such as the triggers for the questionnaires, start and end dates of the study, and so on; and the creation of a customized Android application to distribute among your study participants.

Defining Parameters

Questionnaires and according parameters are defined in using a web interface in the backend server. At the URL

`http://<IP address:Port>/static/QuestionnaireAdmin_en/QuestionnaireAdmin_en.html` you can add, edit or delete questionnaires. The following parameters can be set:

- *Title*: name of the questionnaire as it appears in the list of questionnaires in the application
- *Description*: a more detailed description of the questionnaire
- *Start Date*: the start date in format YYYY-MM-DD HH:MM. The questionnaire is only visible and active within start and end date.
- *Repetition Interval*: the interval after which the questionnaire is shown again (in minutes)
- *Repetition Number*: the total number how often this questionnaire appears. "0" means that there is no limit.
- *Automatic Opening*: "1" means that the questionnaire is triggered based on defined criteria (see later), "0" means that it has to be opened manually.
- *End Date*: the end date in format YYYY-MM-DD HH:MM. The questionnaire is only visible and active within start and end date.
- *Groups*: Multiple surveys, each having a unique set of questionnaires, can be run at the same time by defining groups. The IDs define to which groups the questionnaire belongs (multiple IDs can be entered comma-separated).
- *Log Activity*: the activity names to be observed by the application. Multiple activities can be entered comma-separated.
- *Log Package*: the package names to be observed by the application. Multiple packages can be entered comma-separated. You can also use regular expressions. If "Automatic Opening" is enabled, questionnaires appear automatically after one of the listed activities or packages has been used.

Creating Questionnaires

The questions itself are likewise entered using the web interface. The following types are available:

- *Radio*: a single-choice question (at most 7 choices)
- *Check*: a multiple-choice question (at most 7 choices)
- *Likert*: a Likert scale (at most 7 steps)
- *Dropdown*: a set of more than 7 answers
- *EditText*: a free text answer field
- *ScaleEdit*: a slider and an additional text field for specifying numeric values
- *Text*: text for showing information (e.g. explanatory texts or instructions)

Each question added to the questionnaire will appear on a new screen page. Questions can be rearranged in order by dragging and dropping them to a new position.

The created questionnaire is saved as XML file. Alternatively, the XML file can be created manually according to the following schema:

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="questionnaire" type="questionnaireType"/>

  <xsd:complexType name="questionnaireType">
    <xsd:sequence>
      <xsd:element name="title" type="xsd:string"/>
      <xsd:element name="description" type="xsd:string"/>
      <xsd:element name="startdate" type="xsd:string"/>
      <xsd:element name="repeattime" type="xsd:integer"/>
      <xsd:element name="repeatnumber" type="xsd:integer"/>
      <xsd:element name="autoopen" type="xsd:integer"/>
      <xsd:element name="groupids" type="xsd:string"/>
      <xsd:element name="enddate" type="xsd:string"/>
      <xsd:element name="logactivity" type="xsd:string"/>
      <xsd:element name="logpackage" type="xsd:string"/>
      <xsd:element name="questions" type="questionsType"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="questionsType">
    <xsd:sequence minOccurs="0" maxOccurs="unbounded">
      <xsd:choice>
        <xsd:element name="radio" type="radioType"/>
        <xsd:element name="check" type="checkType"/>
        <xsd:element name="likert" type="likertType"/>
        <xsd:element name="dropdown" type="dropdownType"/>
        <xsd:element name="text" type="textType"/>
        <xsd:element name="edittext" type="edittextType"/>
        <xsd:element name="scaleedit" type="scaleeditType"/>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="radioType">
    <xsd:sequence>
      <xsd:element name="question" type="xsd:string"/>
      <xsd:element name="choice" type="xsd:string" maxOccurs="7"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="checkType">
    <xsd:sequence>
      <xsd:element name="question" type="xsd:string"/>
      <xsd:element name="choice" type="xsd:string" maxOccurs="7"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="likertType">
```

```

        <xsd:sequence>
            <xsd:element name="question" type="xsd:string"/>
            <xsd:element name="choice" type="xsd:string" maxOccurs="7"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="dropdownType">
        <xsd:sequence>
            <xsd:element name="question" type="xsd:string"/>
            <xsd:element name="choice" type="xsd:string" maxOccurs="7"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="textType">
        <xsd:sequence>
            <xsd:element name="question" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="edittextType">
        <xsd:sequence>
            <xsd:element name="question" type="xsd:string"/>
            <xsd:element name="prompt" type="xsd:string" minOccurs="0"/>
            <xsd:element name="numlines" type="xsd:integer" minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="scaleeditType">
        <xsd:sequence>
            <xsd:element name="question" type="xsd:string"/>
            <xsd:element name="value" type="xsd:integer" minOccurs="1"/>
            <xsd:element name="maxValue" type="xsd:integer" minOccurs="1"/>
            <xsd:element name="unit" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>

```

Building the Application

After you have created your questionnaires, you need to build your Android application. Use the Android project that comes in the “Questionnaire” folder in the ZIP file. Move the generated XML files in the assets folder (the file names don’t play a role as long as they have the extension XML).

The next step is the configuration of the Questionnaire service, which you do in the QuestionnaireService.template file. This file is a normal Java file, but it has been renamed to “.template” to remind you that you need to adapt it before use. Rename the file extension to “.java” and make the following settings:

- Change the value of the variable “serverIP” to the IP address and port where your Backend server is running.
- Set the values for murTime, pdTime and slTime. They have the following meaning:
murTime: manual upload reminder time (in minutes) after which the user is notified to upload Questionnaires if the upload mode is set to manual
pdTime: periodic downloader time, indicating the interval in which the service checks for newly available questionnaires
slTime: save log alarm time, indicating the interval after which the user is notified when she has forgotten to answer a questionnaire (in interval mode)
- Set the value of the variable “groupId” to the ID you have specified in the questionnaires. Using this switch, you can create different app versions for between-subjects studies, which then can receive different questionnaires.

- Set the value of the variable “mode” to one of the modes VOLUNTARY, INTERVALTRIGGERED or EVENTTRIGGERED. In interval-triggered mode, questionnaires appear in the defined repetition interval. In event-triggered mode, questionnaires appear based on the log activity and package settings. In voluntary mode, questionnaires do not appear automatically.

You can then build the application package (APK) file using your favorite procedure, e.g. using Eclipse and the Android Developer Tools plugin. The APK is ready to be deployed to your study participants.

Note: Since the APK is not distributed via the Google Play store, study participants will have to allow “installation from unknown sources”.

Collecting the Data

After the APK is installed on a mobile device, the application called “Questionnaire” must be launched once to start the background service. It is recommended to combine this with an initial questionnaire that has to be answered. In the main menu, only questionnaires that are valid between the defined start and end date appear. Depending on the defined repetition number, they might disappear after they have answered, in order to prevent that users answer questionnaires multiple times.

Questionnaires can even be added in the course of the study using the web interface. Given that the start and end interval is valid, they will automatically appear in the application’s main menu, or triggered according to the specified setting. It may take up to one day until new questionnaires are found and included to the application.

In case log activities or packages have been defined, their usage is monitored and written to a log file, which is saved on the SD card. Depending on the setting (accessible via Preference menu), log files are uploaded automatically to the server. If automatic upload is disabled or failed, a reminder pops up that questionnaires should be uploaded using the “Upload” menu entry. Both questionnaire answers and app usage logs are sent to the server. However, automatic uploads are only triggered after a questionnaire has been answered. In order to ensure that all data is received, participants should be instructed to press the “Upload” button at the end of the study, in case there is no final questionnaire.

The uploaded data can be viewed at

http://<IP address:Port>/static/DatabaseAdmin_en/DatabaseAdmin_en.html. The interface contains two sections:

- *Answers*: Shows the questionnaire answers of all users
- *Logs*: Shows the log data of all users

With an input field, the list can be filtered (by any type of data). The available fields are:

- For Questionnaires: a uniquely created user ID, the time of the log entry, the title of the questionnaire and the answers,
- For Logs: The user ID, the time of the log entry, the usage duration of the app, the activity and package names, a “to package name” field, and the location (if a location provider has been enabled on the phone). The “To package name” field indicates the

subsequent package, telling you whether the user changed e.g. to the home screen or to another application.

Evaluating the Data

The evaluation tool is available at

`http://<IP address:Port>/static/Evaluation/Evaluation.html`. This tool further analyzes the collected data to retrieve the information you are really interested in. What comes with this ZIP file is an example that you might need to modify for your individual needs and your individual study.

You might therefore want to change the files “views.py”, “Evaluation.js” and “Evaluation.html”.