Surveys editor guidelines

Introduction

This document contains the guidelines to start using the Influenzanet Surveys Editor. The Surveys Editor is a tool conceived to let non-programmers design, build, test and publish a web survey inside the Django framework of the Influenzanet platform.
The Surveys Editor is multi-language tool deployed as a Django app completely integrated in the Influenzanet Django framework.

Generic surveys

The following issues will be explained:

- *How to access the Surveys Editor*
- *How to create a new survey*
- *How to create questions of different kind (single choice, multiple choice, tables with drop down menus etc)*
- *How to create rules to regulate the logic flow inside a survey*
- *How to save, test and publish a survey*
- *How to import/export an already existing survey*
- *How to translate a survey*

How to access a survey editor

To access the surveys editor, one needs admin permissions. From the Django backend, the editor can be accessed from the pollster app, clicking on the page: “surveys”, where there will be a button “Full Editor” to access the Editor interface. To access this interface directly, one can visit the url: http://example.com/admin/surveys-editor/

How to create a new survey

When accessing the surveys editor interface, you are faced with two possibilities:

1) Browse the local file system for an already existing survey and import it (explained later)
2) Create a new survey using the button “new survey”

Let’s create a new survey. An empty page is created and a title and short name have to be assigned. The short name is important because it will not only be used to reference the survey (e.g. example.com/surveys/{shortname}) but it will be also used as the name of the table (or tables derived from the survey) that will be created in to the DB to store the data coming from this survey.

Once that the survey name and the short name are saved, you can click on the “Full Editor” button. This will show the blank survey-page in which the survey can be edited. The table in the DB has not been created yet, it will be created only once the survey IS PUBLISHED. Until that moment, one can change the shortname, the title, etc etc.

How to create questions of different kind (single choice, multiple choice, tables with drop down menus etc)

The interface now is ready to add questions. On the right side of the blank page, there is a drop down menu to create a "New Question". The user can choose among the following options: Open answer, Single Choice, Multiple Choice, Matrix Select, Matrix Entries.

Before explaining in detail the various options, it must be noticed that when creating a question of any kind, if you click on the question just created, on the right side a panel with many options is opened. These options let you do the fine tuning of the features for each single questions (more details for each option):

- *Data name (name of the column corresponding to this question in the table)*
- *Data Type (PostalCode, Numeric, etc and YearMonth that is used in the Intake for year and month of birth only)*
- *Visual (the kind of options for single choice (radio buttons or drop down menu)*
- *Mandatory (it lets you choose if the question has to be mandatory or not)*
- *Start (lets you choose if the question is visible from the beginning or hidden and triggered by some rule)*
- *Title (title of the question)*
- *Text (some description or a more verbose title)*
- *Tags (can be used to customize the XML tags but in most of the cases it can be ignored)*
- *Regular expressions (regexp can be used to validate user input on submission)*
- *Error Message (the error you want to show when the regexp is not fulfilled).*
- *Add rule (see later).*

Let's go through the different kind of questions you can create:

Option 1) Open answer: creates an open field whose type can be chosen from the right panel. The open field is where a regexp can be used (only if the data type is text, of course).
<u>How it is stored in the DB:</u> the data name gives the name of the column for this question.

Option 2) Single Choice: creates a question with single choice options that can be radio buttons or drop down menus (choose this from the "visual" on the right panel). When you create a single choice question, you have one option only. To add more options, go on the right panel and click on "add choice". To edit the single options, click on each of them and add text, description (it is what will be visualized in the tooltip) and value (the value that will be stored into the DB corresponding to that option). If you select Open answer "yes", you create an open field in correspondence of this option. This open field can be used to enter an input with the same data type described above. NOTE: The data type of this open field can be different from the data type of the whole question and it can be chosen with the option "Open Option Type".
<u>How it is stored into the db:</u> the data name gives the name of the column for this question. When you select an option, the corresponding value is stored in that column field (whose name is Data-

Name). If the single option has an open field, the value is stored in the column whose name is DataName while for the open field, the value is stored in a column whose name is DataName_open.

Option 3) Multiple Choice: creates a question with multiple choices. Again, when you create a multiple choice question, you have one option only. To add more options, go on the right panel and click on “add choice”. To edit the single options, click on each of them and add text (title of the question), description (tooltip text) and Value.
How it is stored into the db: for each option you have a column whose name is DataName_Value (the fields are Boolean). If the single option has an open field, the value is stored in yet another column whose name is DataName_Value_open

Option 4) Matrix Select: This option lets you create a matrix of drop down menus. When you create a matrix select question, you have only one unnamed column and one unnamed row. You can add more columns, rows and choices from the right panel. The data type is the one selected for the whole question.
How it is stored into the db: for each cell of the grid you have a column whose name is: DataName_multi_row{n}_col{m} (where n is the n-th row and m is the m-th column).

Option 5) Matrix entries: This option lets you create a matrix of open fields. When you create a matrix entries question, you have only one unnamed column and one unnamed row. You can add more columns, rows from the right panel. The data type is the one selected for the whole question.
How it is stored into the db: for each cell of the grid you have a column whose name is: DataName_multi_row{n}_col{m} (where n is the n-th row and m is the m-th column).

How to create rules to regulate the logic flow inside a survey

Below the right panel for each question, there’s the “Add rule” widget. This tool lets you regulate the logic flow among the survey questions. It can be used to open a hidden question, to hide a question, regulate the exclusivity among the options of a multiple choice etc. All the rules have one “trigger” question, corresponding to the options in a question that, when selected, trigger the rule. The rule has one target question that can be selected from “Question” drop down menu.
The type of the rule can be chosen from the drop down menu and it has several options (explained below). To add a new rule between two questions, select the trigger question and create a rule.
Sufficient: when two or more rules are of the same type and have the same target you have to choose if only one of them is sufficient to trigger the common target (sufficient=yes) or all of them have to be satisfied to trigger the common target (sufficient = no).

Type Show/hide question: show or hide a question when one of the trigger options is selected. To activate it, select the trigger options of the question you are editing (if the option is an open field, you won’t be presented with any trigger option, the rule will be triggered when the field is filled in) and select the target question from the drop down menu “Question”.

Show/hide options: show or hide an option when one of the trigger options is selected.
To activate it, select the trigger options of the question you are editing (if the option is an open field, you won’t be presented with any trigger option, the rule will be triggered when the field is filled in) and select the target question from the drop down menu “Question”.

Check/uncheck options: check or uncheck an option when one of the trigger options is selected. To activate it, select the trigger options of the question you are editing (if the option is an open field, you won't be presented with any trigger option, the rule will be triggered when the field is filled in) and select the target question from the drop down menu "Question".

Exclusive: this is used to make a choice exclusive in a multiple choice question. The trigger option here is the option that will be exclusive with respect to the others. There is no target question. If I select more than one "exclusive" option, I will have a group of exclusive options that, when one of them is selected, all other options are un-selected (even those belonging to the group).

Future fill: when filling in a survey and answering a trigger question with this type of rule, next time you fill in the questionnaire, the target question will be pre-filled with the value chosen the first time (fill in the future).

Future show/hide question: when filling in a survey and answering a trigger question with this type of rule, next time you fill in the questionnaire, the target question will be hidden/shown.

Future show/hide option: when filling in a survey and answering a trigger question with this type of rule, next time you fill in the questionnaire, the target options will be hidden/shown.

Fill: when filling in a survey and answering a trigger question with this type of rule, the target question is pre-filled with the value chosen the previous time you filled in the survey (fill from the past). When you edit an open field question, the rule widget lets you choose between "rule" and "derived value". The "rule" type works as described above. The derived value is a tool used to create a trigger using an input inserted by the user in the open field. If you select "derived value", an "Empty range" is created. The range refers to a date (if the data type is a date) or a number (if the data type is a number) etc. The superior and inferior limits can be chosen. There is a special derived value for dates of birth ("Years ago"). For example the trigger can be used to show/hide a question for users who entered a date of birth referring to an age less then the inferior limit and greater than the superior one. (e.g. Year ago=[50, 15] corresponds to a question that will be triggered for person older than 50 and younger than 15). If the derived value is set on a question whose data type is text, in the derived value panel, from the "Type" widget you can choose among "Range" and "Regular Expression". If you select "Regular Expression" ("Range is not much of a use for a text), you'll be able to insert a regexp that will appear as a trigger option if you want to set a rule on that question.

A Regular Expression is a concise and flexible means for "matching" (specifying and recognizing) strings of text, such as particular characters, words, or patterns of characters. Abbreviations for "regular expression" include "regex" and "regexp".

To read more:
http://en.wikipedia.org/wiki/Regular_expression
http://docs.python.org/howto/regex.html
http://www.regular-expressions.info/tutorial.html

How to save, test and publish a survey

To save the survey during the editing process, you can use the widget on the right top area of the panel. If you want to test the questions and rules, you can use the tab on the top of the page to test the survey. Once you are happy with your survey, you can publish it by clicking on the tab "All surveys" and then clicking on "publish" in correspondence of the survey you just finished editing.
If you try to publish the survey for the first time, the software performs controls and consistency checks on the questions, so you could be prompted for some errors notifications. Once you have fixed all the inconsistencies reported by the system, you can try again to publish the survey.

!!! NOTE:* *If the publication goes through, a table in the Database is finally created (it's not created until you publish the survey for the first time), with the name pollster_results_ {shortname}. It won't be possible, then, to edit the survey. If you want to change something, you have to unpublish the survey but once you publish it again, a DIFFERENT table will be created, with the same name as the previous one (pollster_results_shortname) and the previous table is moved to the name pollster_results_{shortname}_{timestamp}

How to import/export an already existing survey

From the "Surveys" page, where all the surveys are listed, for each survey you'll find an "export" option. If you use it, the survey definition will be exported to an XML file preserving all the flow and structure. This XML can be imported to another platform to create an identical survey.

How to translate a survey

The default language of the surveys editor is English. The surveys can be built in any language and then translated to any other language. You can use the tab "Translate" to translate the titles and the options text. When a user visits the page for a survey, he/she will be prompted with the text corresponding to the published translation found for the current Django language (or the default text if the language is not specified).

Special Surveys: Intake and Weekly

The Influenzanet platform has two special surveys that are treated slightly differently in terms of data storage in to the DB. These surveys are described by the Gold Standard Questionnaires document: http://www.influenzanet.eu/media/cms_page_media/5/110901-Questionnaires_UK.pdf

The short names for these surveys are mandatory: intake and weekly.
These short names have to be used because the rest of the platform uses them to select and use information about users and persons (e.g. to manage the example.com/survey/people page or when a user registers, he is sent to the page example.com/survey where as default he has to fill in the intake and after that the weekly).

To add a map centered on the user’s postal code in the Weekly Questionnaire response page

In templates/survey/thanks.html a map is pre-allocated, so you only need to create a Weekly Questionnaire chart with shortname “neighbourhood” and selecting the type “Centered Map”.
The box where the map will be shown is regulated by the following HTML:

```
<div class=”survey-chart”>
   <div class=”chart” data-chart-url=”{% url pollster_chart_data “weekly” “neighbourhood”  %}” data-chart-type=”google-map-centered” style=”width:100%;height:330px”>
   </div>
</div>
```

Javascript at the beginning of the template will automatically initialize and show the chart.

To visit the page for each publish survey

Use the url: **http://example.com/surveys/SHORTNAME**

where SHORTNAME is the shortname created for each survey.

Maps and Charts

It is possible to extract graphical information from the surveys data. In the “Surveys” page, close to each survey name, there’s the possibility to select the “charts” page containing the list of charts (and maps) created for this specific survey. To create a new one, insert a short name in the open field and click on create chart (also if you want to create a map). When you click on “create chart”, you are prompted with a grey area (where your chart/map will appear) and a panel on the right part of the page. From the Type option you can choose among: Google Chart, Google Map, Google Map (centered on user). Charts and maps are accessible as Django-CMS plugins. Once a chart or a map is created, it can be published by using the Django plugins in each page
Everytime a chart is created, in the DB a nee table is created with the name: pollster_results_SURVEYNAME_CHARTNAME

Google Chart:

By using the “edit” option close to the Google Chart selection (this option will appear as soon as you have available data to be displayed, i.e. as soon as you start filling in the corresponding survey), you can choose the kind of chart you want to create: scatter plot, bar plot, box plot, etc (all the different options offered by the Google Charts API).
You can then use the “Sql source query” to select the data you want to display. Here you have to use the actual names for the tables generated by the surveys in the DB.

For example, if you created a survey with the short name “foo”, you’ll find your data in the table pollster_results_foo and you’ll have to perform sql queries on this table.

Example: To count the number of “foo” surveys filled in by each person:
First select a Google Chart bar plot and then write the the following in the “SQL source query”

```
SELECT global_id, count (*) FROM pollster_results_foo GROUP BY global_id
```

In the grey area, you’ll have a bar plot showing the number of “foo” questionnaires filled in by each user. To display results filtered by user or person or none of the two, you can use “Results filter”. In the former case, you must ensure that the “user” column is returned in the SELECT clause. In the second case, you have to SELECT both “user” and “global_id” columns.

Another example: To count the number of men and women who have filled in the questionnaire, you can perform the SQL query (if “Q1” is the column corresponding to the intake question where the gender is asked):

```
SELECT
‘male’ as gender, count(*) as num
FROM pollster_results_intake
WHERE “Q1”=’0’
UNION
SELECT ‘female’ as gender, count(*) as num
FROM pollster_results_intake
WHERE “Q1”=’1’;
```

This query will produce the result in the DB:

```
gender| num
--------- + -----
female | XXX
male   | YYY
```

And you’ll be able to see this result in the grey interface in the form of a table or a bar plot or whatever you choose.

***!!! NOTE:* when you perform a query, be sure you are always selecting the correct table. For example, if you have modified the intake questionnaire after the first publication and you want to create a chart, in principle you should to perform the query on the most recent table: pollster_results_intake_TIMESTAMP.**

It is also possible to merge by hand the most recent table into the original pollster_results_intake and use the latter to perform the query. If the table structure into the DB hasn’t been altered (no changes in “Data Name” o changes in the options values o data types) a simple insert is enough:

```
INSERT INTO pollster_results_intake
FROM SELECT * FROM pollster_results_intake_20111020_1228;
```

In case the table structure has been altered, the changed columns have to be explicitly inserted, for example:

```
INSERT INTO pollster_results_intake (Q1, Q2, Q3_0, Q3_1, Q3_1_open_data)
FROM SELECT Q1, Q2, Q3_0, Q3_1, Q3_1_open_data
FROM pollster_results_intake_20111020_1228;
```

For charts created before the modification, it is recommended the use of the command (from the platform installation): source ./bin/activate && python manage.py chart_update

It is usefult to have a cron script like the following:

```
#!/bin/bash
set -e
cd ${PATH_DELL_INSTALLAZIONE}
source ./bin/activate
python manage.py chart_update
```

For the Weekly questionnaire, at the end of the survey each user receives a response about his/her symptoms and can see the “history” of his/her surveys. The table from which these results are visualized is automatically updated in case something is modified in the survey.

Google Map:

If you use this option, you’ll be able to create a google map with tiles containing information retrieved from the DB.

Example: to count the number of users who have filled in the intake questionnaire in each postal code and display them on a Google map:

```
SELECT
“QN3”::text AS zip_code_key,
‘#ff0000’::text AS color,
count(*) AS “Count”
FROM
pollster_results_intake
GROUP BY 1,2
```

Note: you have to have preloaded shape files for the postal codes. See instructions below.
Note2: for the Google Maps, the zip_code_key and color columns are mandatory. All the other selected columns will be displayed in the tooltip in the form: column name: value.

In the SQL Source Query field, all the valid PostreSQL syntax can be used: you have full access to the Postgres DB (even DELETE and INSERT so be extremely CAREFUL!!).

To import shapefiles:

1) You’ll have already installed and enabled PostgreSQL PostGIS extensions

2) From inside the directory containing all the GIS files, import raw shapefile (.shp) into a temporary table:

```
shp2pgsql -s 4326 -c ita_cap_region.shp zip_temp > zip_temp.sql
psql -h localhost -p 5432 -U dbuser dbname -f zip_temp.sql
```

3) Replace PostgreSQL parameters with the ones used in the Django settings and make sure your data uses the common WGS84 projection (SRID:4326)

4) Have a look at the generated table and import it into real table:

```
$ psql -h localhost -p 5432 -U dbuser dbname
> \d zip_temp
... (some output here, look for the zip and geometry column names) ...
> INSERT INTO pollster_zip_codes
(country, zip_code_key, geometry)
SELECT ‘IT’, cap, the_geom FROM zip_temp;
> DROP TABLE zip_temp;
```

Note how we set the country to ‘IT’ for Italian zip (CAP) codes. For other countries, you’ll have to use the corresponding country code.

5) To test the data just create a “Google Map” chart on any survey (the test doesn’t use survey data so even an empty survey is OK) and use the following query:

```
SELECT zip_code_key, ‘#ff0000’::text AS color
FROM pollster_zip_codes
```

The first time this is slow, because draws ALL the available geometry.

Setting the size for each map:

The size of each map is set in /media/base/css/default.css:

```
.survey-chart .chart {
 width: 440px; height: 330px; position: relative;
}
```

Each chart has a class:

```
chart-{{ chart.survey.shortname }}-{{ chart.shortname }}
```

(for example “chart-weekly-neighbourhood”), so from your CSS you can add new rules to overwrite the previous:

```
.survey-chart .chart.chart-weekly-neighbourhood {
 width: 220px; height: 165px;
}
```