

---

# **DDI on Rails Documentation**

***Release 1.0b***

**Marcel Hebing**

**Apr 11, 2018**



---

## Contents

---

<b>1</b>	<b>Contents</b>	<b>3</b>
1.1	User guide . . . . .	3
1.2	Data management and documentation . . . . .	4
1.3	Imports and exports . . . . .	5
1.4	API . . . . .	13



---

**Note:** This is still work in progress. Please visit the official documentation on <http://www.ddionrails.org>

---



## 1.1 User guide

---

**Note:** This is a draft version.

---

This is a guide how to use the data portal DDI on Rails, which builds the foundation for the new version of SOEPinfo. DDI on Rails is created in order to assist the user to explore survey data (e.g. the SOEP), to compile personalized datasets and to publish the results on the publication database. Primarily, it guides the user throughout the entire process of their research projects using the SOEP data from conception to publication and citation.

### 1.1.1 Search engine

On the main page a search option can be found, providing a quick way to be navigated directly to the respective results if the user has a specific variable, dataset, or topic in mind. It also helps for a quick overview if the user's interest is to find out if there is any data available for his/her research topic/question.

For instance, if you are looking for at the results for the keyword "age", you can find a total of 2614 items listed. However, this might be overwhelming. Thus, it provides you a way to narrow your search output, using so-called facets. Possible classes above the total results are "Concepts", "Variables", "Questions" and "Publications" - each of them with optimized facets for the particular class.

The select box for studies is the only one, available in all views. This makes it easy for the data users of one particular study to consistently adjust all result views. The descriptions to those studies can be found on the very top of the page under "Studies" after a particular study is selected and will be specified in the next section again.

The already mentioned classes with its assigned symbols are listed again underneath the "Study" option and below that another sorting option regarding the analysis unit can be selected.

### 1.1.2 Studies

DDI on Rails incorporates various SOEP-studies, including “SOEP Core study”, “SOEPlong”, “Families in Germany”, “SOEP Innovation Sample”, “SOEP Pretest”, “Base II”, and “SOEP Test Study”. A particular study can be accessed as shown in the picture below. After a study of interest is selected, a general overview will be displayed. Moreover, the total number of involved variables and datasets can be viewed.

In the variable browser, there is another search engine adapted to the chosen study. That way you can find your required variables in the particular study. The same holds for the dataset browser. Furthermore, the variables can be selected according to the desired analysis unit or period on the left-hand side.

### 1.1.3 Topics and concepts

A variety of topics can be selected on the very top of every page next to “Studies”. After clicking on a particular topic, several subtopics appear. The user may also use the search engine to look for specific concepts regarding the chosen topic.

Concepts are used to group variables within one study or in between multiple studies, that might have slight variation but still put themselves forward to be analysed in a comparative way. They replace the so-called “item correspondence” from the former SOEPinfo.

### 1.1.4 Publications

Under “publications” you can search for any keyword and you will be directed to a list of papers that involve the searched word(s). Each result provides a link to the publication for direct access.

### 1.1.5 Workspace and basket

While the former SOEPinfo allowed the use of the basket and its script generator without any login information, the new system requires you to log in in order to create baskets for variables.

The login is necessary to enable some of the new features in DDI on Rails, among others the possibility to store multiple baskets at a time and access those baskets directly from most statistical packages.

After having signed in, a new basket-symbol appears on the very top. Now you can create new baskets and fill in the individual information for each basket. To enable comparison between studies and distributions (versions of one study) later in the process, it is necessary to bind the basket to a particular distribution.

All variables of interest accessed as previously explained can be selected into your basket now by clicking on the green “Add to basket” symbol underneath the variable in the variable browser. The number next to the basket indicates how many variables are already integrated into your basket.

## 1.2 Data management and documentation

---

**Note:** Plans for this chapter:

- Conventions and concepts for data management
  - Integrating metadata generation and data management in a metadata-driven process.
  - Lifecycle model for data mangement, documentation, and re-use.
-

## 1.3 Imports and exports

---

**Note:** Attention: the import procedure is about to change in the next version of DDI on Rails.

---

### 1.3.1 Import formats

#### CSV formats

---

**Note:** The full list of CSV imports is currently documented on [ddionrails.org/imports](http://ddionrails.org/imports)

---

#### Markdown

In most imports, there is a `description` field using Markdown. For more information about the Markdown markup language, see: [Daring Fireball](#).

#### Conventions

- Some fields in the CSV exports are not part of the import. Those fields start with `view_` variables and datasets.
- Columns with the `internal_` prefix are indented for internal use only and will not be imported (e.g. `internal_comment`).
- Language codes, for all translation purposes: [ISO 639-1](#)

#### XML formats

- **Endnote:** Publications in Endnote's XML export.
- **r2ddi:** DDI-Codebook-based XML, generated by `r2ddi`.
- **QeDML:** QeDML-XML from QLILB.

#### Other formats

- **EndnoteKeys** are a special import of two columns (accession number and keywords) in addition to the normal XML import. Endnote exports everything but the keywords to XML what makes this import necessary.

### 1.3.2 Import structure

#### Top level

```
import/
| system/      # -> system-wide imports
| study-first/ # \
| study-second/ # }-> one folder per study
| study-third/  # /
```

## All levels

```
import/  
|  system/  
|  |  endnote.xml  
|  |  endnote-keys.txt  
|  |  ddiOnRails.png  
|  study-first/  
|  |  studies.csv  
|  |  variables.csv  
|  |  ...all other csv files...  
|  |  files/  
|  |  |  ...all files for public folder...  
|  |  qedml/  
|  |  |  ...questionnaires in QeDML-XML-format...  
|  |  r2ddi/  
|  |  |  version/  
|  |  |  |  ...dataset descriptions in DDI-C-XML...  
|  study-second/  
|  |  ...like study-first...
```

### 1.3.3 CSV Imports

#### studies.csv

##### Columns

**organization** Name of the organization (foreign key)

**study** Name of the study (primary key).

**label** Human-readable label.

**description** Description (using Markdown).

**html\_description** HTML description (DEPRECATED).

**language\_string** Whitespace separated list of languages used in the study as two-figure language codes (e.g. “de en”). These parameters are used to import and export the translations of questionnaires and datasets.

**import\_url** URL from where all import files are retrieved.

**files\_url** URL from where files are loaded interactively.

**import\_config** Additional import parameters, currently not used.

#### topics.csv

##### Columns

**topic** Name of the topic (primary key).

**parent** Name of the parent topic (foreign key). If empty, this topic becomes a root-level topic, requiring an icon.

**label** Short label.

**description** Description using Markdown.

### concepts.csv

#### Columns

**concept** Name of the concept (primary key).

**topic** Name of the topic (primary key).

**label** Short label.

**description** Description using Markdown.

### periods.csv

#### Columns

**period** Name of the period (primary key).

**label** Short label.

**description** Description using Markdown.

### analysis\_units.csv

#### Columns

**analysis\_unit [PK]** Name of the analysis unit.

**label** Short label.

**description** Description using Markdown.

### conceptual\_datasets.csv

#### Columns

**conceptual\_dataset** Name of the conceptual dataset (primary key).

**label** Short label.

**description** Description using Markdown.

### logical\_datasets.csv

#### Columns

**study** Name of the study (primary key).

**logical\_dataset** Name of the dataset (primary key).

**label** Short label.

**description** Description using Markdown.

**conceptual\_dataset** Name of the conceptual dataset (foreign key).

**analysis\_unit** Name of the analysis unit (foreign key).

**period** Name of the time period (foreign key).

### logical\_variables.csv

#### List of Columns

**study** Primary key, name of the study.

**logical\_dataset** Primary key, name of the dataset.

**logical\_variable** Primary key, name of the variable.

**label** Human-readable label.

**concept** Name of the underlying concept, foreign key to concepts.csv.

**questionnaire** Name of the underlying questionnaire, foreign key to questionns.csv.

**question** Name of the underlying question, foreign key to questions.csv.

**item** Name of the underlying item, foreign key to questions.csv.

**is\_primary\_key** Boolean indicator, if this variable is part of the dataset's primary key.

**basket\_key** Name of an study-specific identifier in this dataset, which is used for the script generator.

**basket\_is\_default** Boolean indicator, whether a script generator should include this variable by default, if its dataset is used.

#### Special Rules

- The link to a question (or question item) is only established if the question already exists. There are no new questions created by variables.csv.

### distributions.csv

#### Columns

**study** Name of the study (primary key).

**distribution** Name of the Distribution (primary key).

**label** Short label.

**description** Description using Markdown.

**active**

Boolean value ("true" or "false"), indicating whether this is currently the active distribution of the study.

### **datasets\_distributions.csv**

#### **Columns**

**study** Name of the study (primary key).

**distribution** Name of the distribution (primary key).

**dataset** Name of the dataset (primary key).

**version** Versio of the dataset (primary key).

This table builds a has-and-belongs-to-many relationship between datasets and distributions. Thus, it only consists of key values without any attributes.

### **variables.csv**

This Format is **export only**.

#### **List of Columns**

**study** Name of the study (primary key)

**dataset** Name of the dataset (primary key)

**version** Version of the dataset (primary key)

**variable** Name of the variable (primary key)

**label** Short label.

**categories** List of categories in pseudo-JSON format.

**label\_xx & categories\_xx** Translated labels for variables and categories.

### **variable\_categories.csv**

This Format is **export only**.

#### **List of Columns**

**study** Name of the study (primary key).

**dataset** Name of the dataset (primary key).

**version** Version of the dataset (primary key).

**variable** Name of the variable (primary key).

**value** Value of the category (primary key).

**label** Category label.

**frequency** Frequency.

**label\_xx** Translated labels.

## generations.csv

### Columns

**output\_study** Name of the output variable's study (primary key).  
**output\_dataset** Name of the output variable's dataset (primary key).  
**output\_version** Name of the output variable's dataset version (primary key).  
**output\_variable** Name of the output variable (primary key).  
**input\_study** Name of the input variable's study (primary key).  
**input\_dataset** Name of the input variable's dataset (primary key).  
**input\_version** Name of the input variable's dataset version (primary key).  
**input\_variable** Name of the input variable (primary key).

## questionnaires.csv

### Columns

**study** Name of the study (primary key).  
**questionnaire** Name of the questionnaire (primary key).  
**label** Human-readable label.  
**description** Description using Markdown.  
**analysis\_unit** Name of the analysis unit (foreign key).  
**period** Name of the time period (foreign key).  
**dataset** Name of the dataset (foreign key).

## question.csv

### List of columns

(1) Identifier: The first four columns identify are question. Please note that a question can consist of multiple items. In this case the first item is considered to be the root element and the item is either empty or "root".

**study** Name of the study (primary key).  
**questionnaire** Name of the questionnaire (primary key).  
**question** Name of the question (primary key).  
**item** Number of the question item (primary key). If the `item` is empty, the question is considered to be a "root question", which might have items.

(2) Content: The following columns represent the content of a question or item

**number** Question number (integer), as a reference to the position in the questionnaire.  
**text** Question text.  
**instruction** Interviewer instruction.

**answer\_list** Name of the list of answers (foreign key). The `answers.csv`.

**scale** Scale (see list of scales below) of the answers.

**filter** Incoming filters (see definition below).

**goto** Outgoing filters (see definition below).

**label** Label (DEPRECATED).

**description** Human readable description including additional unstructured information.

**concept** Name of question's concept (foreign key). In DDI on Rails the primary link from a question to one or multiple concepts is through the question's logical variables. Nevertheless, it is possible to link a question or an item directly to a concept.

(3) Links to logical variables and concepts (import only): A question can be linked to multiple logical variables. Therefore, DDI on Rails stores this link with the logical variables. Yet, the questions import allows to link every question to one logical variable.

**logical\_dataset** Logical dataset name (foreign key).

**logical\_variable** Logical variable name (foreign key).

(4) Export only: There is a couple of columns that is included in the export but will not be imported.

**view\_sort\_id** Sort order of the questions. The `view_sort_id` is generated from the order of the questions in the import file.

**view\_lft** and **view\_rgt** Export only.

**view\_import\_note** Export only (DEPRECATED).

**view\_first\_concept** Concept of the question, based on the first related variable.

**view\_import\_typ** Export only (DEPRECATED).

**view\_calculated\_number** and **view\_calculated\_item** Special information for imports following the SOEP-QLIB-conventions.

**logical\_variable** Name of the resulting variable (foreign key, import only).

**logical\_dataset** Name of the dataset of the resulting variable (foreign key, import only).

(5) Namespaces (neither imported nor exported): Every study can add an arbitrary number of columns to store additional information that are not intended to be imported in DDI on Rails. Those columns are prefixed with `internal_`

## Scales

**txt** Only display the text, no variables are generated. All filters and instructions still apply.

**chr** Result is a character string.

**int** Result is a integer.

**dec** Result is a number with decimals.

**bin** Result is either true, false (equals "null")

**cat** Result is a pre-defined answer category. See `answer_list` for possible answers.

### Rules for filter and goto

Filter and goto definitions consist of question names and symbols only, no keywords (e.g. “goto”) are used.

- Symbols ( ) = < > @ | & : != <= >=
- Filter (AGE > 20) & (SEX = 1) means: this question is asked if “age” is greater than 20 and “sex” is 1
- Goto (2 @ TARGET) means: if the answer to the current question is 2 then go to question “target”
- Refer to items using the colon as a separator, e.g. (PSOR:2 = 3).
- Value lists and ranges: (x = 1:3) is equal to (x = 1, 2, 3) is equal to (x = 1) | (x = 2) | (x = 3)

### answers.csv

#### List of columns

**(1) Identifiers:** The first three columns identify an answer list. An answer list always refers to one questionnaire. It is not possible to refer from a question in questionnaire A to an answer list in questionnaire B.

The fourth column (value) identifies an item of an answer list.

**study** Name of the study (primary key).

**questionnaire** Name of the questionnaire (primary key).

**answer\_list** Name of the answer\_list within the questionnaire (primary key).

**value** Integer value of the answer (primary key).

**(2) Content:** The content of an item is a label. This label can be translated.

**label** Answer label in the primary language (usually English).

**label\_\*** Translations of the label. Please replace \* by a two-digit language code, e.g. label\_de for a German label.

### Features

Answer labels are translatable. The language of the translation is set using a two letter code, e.g. label\_de for a German label. The default language for the column label is English.

### translations.csv

Please keep in mind that translations.csv is only an export format. The import of translations is part of the respective translatable object.

The term “translatable” refers to an object that has one or more attributes that can be translated.

### Columns

**class** Class of the translatable.

**id** ID of the translatable.

**attribute** Translated attribute of the translatable.

**text** Original version of the text.

**language** Language of the translation.

**translation** Translated version of the text.

### script\_generators.csv

#### Columns

**study** Name of the study (primary key).

**distribution** Name of the distribution (primary key).

**script\_generator** Name of the script generator's class (primary key). Note: Please be mindful of case sensitivity.

**label** Short label.

**description** Description using Markdown.

## 1.4 API

### 1.4.1 Basket API

Every user can have multiple baskets, where variables can be stored.

#### Basket

Returns a basket instance. The owner of the basket must be logged in.

```
/baskets/:id
```

#### Ressource Properties

A basket is represented by the following properties:

Property	Type
id	int
basket_name	String
variable_list	Collection<Variable>
owner	User
study	String

Supported HTTP Methods: GET, PUT, DELETE,

Optional Parameters: None.

## Basket List

Returns a list of all baskets belonging to the currently logged in user.

```
/baskets
```

Supported HTTP Methods: GET, POST

Optional Parameters: None

## Variables List of a Basket

Returns a list of Variables associated with the specified basket.

```
/baskets/:id/variables
```

Supported HTTP Methods: GET, POST

Optional Parameters: None

## Remove Variable from Basket

Removes the specified variable from the specified basket.

```
/baskets/:id/variables/:id
```

Supported HTTP Methods: DELETE

Optional Parameters: None

## 1.4.2 Concept API

A concepts represents a ???.

### Concept

Represents a single concept instance.

```
/concepts/:id
```

## Ressource Properties

A concept is represented by the following properties:

Property	Type
id	int
concept_name	String
label	String
?	?

Supported HTTP Methods: GET

Optional Parameters: None.

## Concept List

Represents a list of all concepts.

```
/concepts
```

Supported HTTP methods: GET

Optional Paramters: None

## Variables by concept

Get all variables with specified concept.

```
/concepts/:id/variables
```

Supported HTTP methods: GET

Optional Paramters: None

## 1.4.3 Dataset API

A dataset instance is a collection of variables. A dataset is always part of a study.

### Dataset

Represents a single dataset instance.

```
/datasets/:id
```

## Ressource Properties

A dataset is represented by the following properties:

Property	Type
id	int
dataset_name	String
variables	Collection<Variable>
?	?

Supported HTTP Methods: GET

Optional Parameters: None.

## Dataset List

Represents a list of all datasets.

```
/datasets
```

Supported HTTP methods: GET

Optional Paramters: None

## 1.4.4 Study API

A study instance is a collection of datasets.

### Study

Represents a single study instance.

```
/studies/:id
```

### Ressource Properties

A study is represented by the following properties:

Property	Type
id	String
study_name	String
datasets	Collection<Dataset>
?	?

Supported HTTP Methods: GET

Optional Parameters: None.

### Study List

Represents a list of all studies.

```
/studies
```

Supported HTTP methods: GET

Optional Paramters: None

### Included Datasets

```
/studies/:id/datasets
```

Returns a list of all datasets associated with the specified study.

Supported HTTP methods: GET

Optional Paramters: None

### Included Variables by Dataset

```
/studies/:id/datasets/:id/variables
```

Returns a list of all variables included in specified dataset and study.

Supported HTTP methods: GET

Optional Paramters: None

## Included Variables

```
/studies/:id/datasets/variables
```

Returns a list of all variables included in specified study.

Supported HTTP methods: GET

Optional Parameters: None

## 1.4.5 User API

A user instance represents a person who has registered with ddionrails.

### User

Represents a single user instance.

```
/users/:id
```

### Ressource Properties

A user is represented by the following properties:

Property	Type
id	int
email	String
is_active	boolean
date_joined	timestamp
username	string

Supported HTTP Methods: GET

Optional Parameters: None.

### User List

Represents a list of all users.

```
/users
```

Supported HTTP methods: GET

Optional Parameters: None

## 1.4.6 Variable API

A variable instance represents a ???.

## Variable

Represents a single variable instance.

```
/variables/:id
```

## Ressource Properties

A variable is represented by the following properties:

Property	Type
id	int
variable_name	String
dataset	Dataset
study	Study
analysis_unit	String
boost	int
label	String
label_de	String
period	int
sub_type	String
namespace	String
...	...

Supported HTTP Methods: GET

Optional Parameters: None.

## Variable List

Represents a list of all variables.

```
/variables
```

Supported HTTP Methods: GET

Optional Paramters:

Parameter	Values	Description
dataset	dataset name	Variables included in specified dataset.
basket	basket id	Variables included in specified basket.