

# Wepi Webservice Export

## Introduction

Ce webservice sert à une application tiers pour récupérer les données d'un questionnaire Wepi. Précisément il donne accès aux exports de l'onglet "Afficher les données" ("View data").



The screenshot shows a web interface with three tabs: "Afficher les Données", "Partager le Questionnaire", and "Statistiques". Below the tabs is a table with the following data:

sick	begin	Temperature	diarrhea	Vomiting	Abdominal pain
yes		non	non	non	unsure
yes	09/22/2014				
	09/17/2014	non			
yes	05/09/2014	non	yes	yes	yes

An "Export" dropdown menu is open, showing options: STATA, EpiData, CSV (Euro), CSV (US/UK), CSV (UTF-8), and Structure.

## Description technique

Le webservice est de type REST, seule la méthode GET est implémentée. Il nécessite une authentification par login / mot de passe (authentification HTTP de type "Basic realm").

Adresse du webservice: <https://www.wepi.org/ws/export>

Renvoie la liste des questionnaires de l'utilisateur authentifié, avec pour chaque questionnaire, son nom, et son identifiant. L'identifiant sera utilisé par la suite pour la récupération des données.

```
<response>
  <projects>
    <project>
      <name>Questionnaire flu in Belgium</name>
      <id>502137a827292</id>
    </project>
    <project>
      <name>June 2010 iMove Meeting followup questionnaire </name>
      <id>502a7881f37d1</id>
    </project>
  </projects>
</response>
```

Récupération des données: [https://www.wepi.org/ws/export/id/502a7881f37d1/type/csv\\_euro](https://www.wepi.org/ws/export/id/502a7881f37d1/type/csv_euro)

Les paramètres attendus:

- id: correspond à l'identifiant du questionnaire
- type: format des données voulues (optionnel, par défaut csv\_euro)

Valeurs possibles:

- csv\_euro: csv avec séparateur point-virgule, encodé en ISO-8859-1 (latin1)
- csv\_us\_uk: csv avec séparateur virgule, encodé en ISO-8859-1 (latin1)
- csv\_utf8: csv avec séparateur virgule, encodé en UTF-8
- stata: zip contenant un csv (.stata) et un fichier .do pour importation dans Stata
- epidata: fichier rec lisible par EpiData
- structure: fichier HTML décrivant la structure du questionnaire (liste des variables et des dictionnaires)

La réponse du webservice contient:

- type: type demandé
- content-type: descriptif des données (content-type HTTP)
- hash: clé de hashage des données pour contrôler qu'il n'y a pas eu d'altération des données pendant le transfert
- content: données encodée en base64.

```
<response>
  <type>csv_euro</type>
  <content-type>text/csv; charset=ISO-8859-1</content-type>
  <hash>fddceb3a72071e8c042ba16b423dd80e</hash>
  <content>ImcxbG5hbWUiOyJnMmZuYW...</content>
</response>
```

## Exemple PHP

### Récupération de la liste des questionnaires

```
$sLogin = 'john.doe@gmail.com';
$sPassword = 'secret';
$sUrl = sprintf(
    'https://%s:%s@www.wepi.org/ws/export',
    urlencode( $sLogin ),
    urlencode( $sPassword )
);

$oClient = new Zend_Rest_Client( $sUrl );

/* @var $oResult Zend_Rest_Client_Result */
$oResult = $oClient->get();

// Throw exception if WS return an error
```

```

if ( false === $oResult->isSuccess() ) {
    echo 'An error occured while calling webservice: ' .
$oResult->message();
    exit();
}

/* @var $oXmlResponse SimpleXMLElement */
$oXmlResponse = $oResult->response;
header( 'content-type: text/xml' );
echo $oXmlResponse->asXML();

```

## Récupération des données

```

$sLogin = 'john.doe@gmail.com';
$password = 'secret';
$url = sprintf(
    'https://%s:%s@www.wepi.org/ws/export/id/502a7881f37d1/type/stata',
    urlencode( $sLogin ),
    urlencode( $password )
);

$client = new Zend_Rest_Client( $url );

/* @var $oResult Zend_Rest_Client_Result */
$oResult = $client->get();

// Throw exception if WS return an error
if ( false === $oResult->isSuccess() ) {
    echo 'An error occured while calling webservice: ' .
$oResult->message();
    exit();
}

/* @var $oXmlResponse SimpleXMLElement */
$oXmlResponse = $oResult->response;
header( 'content-type: text/xml' );
echo $oXmlResponse->asXML();

```

# Wepi Export Web service

## Introduction

This web service is for other applications to be able to access a Wepi account and extract the data records from a given questionnaire. To be more precise, it gives access to the “export” function in the “View Data” tab.

TIAC fini

Number of records: 39  
<< first < prev 12 next > last >> 20

Action	NOM, Prénom	telephone	sex:	ddn	symptomes	Date de début	températurec	Diarrhée	Vomis
	Alexandre	01212314564	Masculin	09/03/1980	Oui	09/01/2014	41.0000000	Oui	Oui

Export

- STATA
- EpiData
- CSV (Euro)
- CSV (US/UK)
- CSV (UTF-8)
- Structure

## Technical description

It is a REST web service which uses only the GET method. It requires authentication by username/password (authentication HTTP type “Basic realm”).

Webservice address: <https://www.wepi.org/ws/export>

This sends back the list of questionnaire of the authenticated user with the name and id of each of the account’s forms (questionnaires). The id is used in the next request to get the data.

```
<response>
  <projects>
    <project>
      <name>Questionnaire flu in Belgium</name>
      <id>502137a827292</id>
    </project>
    <project>
```

```
        <name>June 2010 iMove Meeting followup questionnaire </name>
        <id>502a7881f37d1</id>
    </project>
</projects>
</response>
```

Extracting the data: [https://www.wepi.org/ws/export/id/502a7881f37d1/type/csv\\_euro](https://www.wepi.org/ws/export/id/502a7881f37d1/type/csv_euro)

The expected parameters:

- id: corresponds to the id of the questionnaire
- type: data format desired (optional: by default is csv\_euro)

Possible values:

- csv\_euro: csv with semi-colon as delimiter and encoded in ISO-8859-1 (latin1)
- csv\_us\_uk: csv with comma delimiter and encoded in ISO-8859-1 (latin1)
- csv\_utf8: csv with comma delimiter and encoded in UTF-8
- stata: zipped files containing a data csv (.stata) and a .do file for Stata import
- epidata: is a .rec file readable by the application EpiData
- structure: an HTML file describing the data structure (variables and dictionaries)

The webservice response contains:

- type: type of request
- content-type: data description (HTTP content type)
- hash: a hash key for the data to verify that it was correctly transferred
- content: data encoded in Base64

```
<response>
  <type>csv_euro</type>
  <content-type>text/csv; charset=ISO-8859-1</content-type>
  <hash>fddceb3a72071e8c042ba16b423dd80e</hash>
  <content>ImcxbG5hbWUiOyJnMmZuYW...</content>
</response>
```

## PHP Example

Retrieve the list of questionnaires

```
$sLogin = 'john.doe@gmail.com';
$sPassword = 'secret';
```

```

$sUrl = sprintf(
    'https://%s:%s@www.wepi.org/ws/export',
    urlencode( $sLogin ),
    urlencode( $sPassword )
);

$oClient = new Zend_Rest_Client( $sUrl );

/* @var $oResult Zend_Rest_Client_Result */
$oResult = $oClient->get();

// Throw exception if WS returns an error
if ( false === $oResult->isSuccess() ) {
    echo 'An error occurred while calling webservice: ' .
    $oResult->message();
    exit();
}

/* @var $oXmlResponse SimpleXMLElement */
$oXmlResponse = $oResult->response;
header( 'content-type: text/xml' );
echo $oXmlResponse->asXML();

```

## Retrieve the data from a questionnaire

```

$sLogin = 'john.doe@gmail.com';
$sPassword = 'secret';
$sUrl = sprintf(
    'https://%s:%s@www.wepi.org/ws/export/id/502a7881f37d1/type/stata',
    urlencode( $sLogin ),
    urlencode( $sPassword )
);

$oClient = new Zend_Rest_Client( $sUrl );

/* @var $oResult Zend_Rest_Client_Result */
$oResult = $oClient->get();

// Throw exception if WS return an error
if ( false === $oResult->isSuccess() ) {
    echo 'An error occurred while calling webservice: ' .
    $oResult->message();
    exit();
}

```

```
}  
  
/* @var $oXmlResponse SimpleXMLElement */  
$oXmlResponse = $oResult->response;  
header( 'content-type: text/xml' );  
echo $oXmlResponse->asXML();
```