

# JavaDoc Reference

**MITWIRKENDE**

	<i>TITEL :</i> JavaDoc Reference		
<i>AKTION</i>	<i>NAME</i>	<i>DATUM</i>	<i>UNTERSCHRIFT</i>
VERFASST DURCH		14. März 2013	

**VERSIONSGESCHICHTE**

NUMMER	DATUM	BESCHREIBUNG	NAME

# Inhaltsverzeichnis

<b>1</b>	<b>Paket com.frentix</b>	<b>1</b>
1.1	Klasse Resttool . . . . .	1
1.1.1	Syntax . . . . .	1
1.1.2	printUsage() . . . . .	2
<b>2</b>	<b>Paket com.frentix.restapi</b>	<b>3</b>
2.1	Klasse RestConnection . . . . .	3
2.1.1	Syntax . . . . .	3
2.1.2	RestConnection() . . . . .	5
2.1.3	RestConnection(String, int, String, String, String, String) . . . . .	6
2.1.4	RestConnection(URL) . . . . .	6
2.1.5	addEntity(HttpEntityEnclosingRequestBase, NameValuePair...) . . . . .	6
2.1.6	addJsonEntity(HttpEntityEnclosingRequestBase, Object) . . . . .	6
2.1.7	addMultipart(HttpEntityEnclosingRequestBase, String, File) . . . . .	7
2.1.8	applyProperties(Properties) . . . . .	7
2.1.9	createDelete(URL, String, boolean) . . . . .	7
2.1.10	createGet(URL, String, boolean) . . . . .	8
2.1.11	createPost(URL, String, boolean) . . . . .	8
2.1.12	createPut(URL, String, boolean) . . . . .	8
2.1.13	createPut(URL, String, String, boolean) . . . . .	8
2.1.14	execute(HttpUriRequest) . . . . .	9
2.1.15	get(URL, Class<T>) . . . . .	9
2.1.16	getBaseURI() . . . . .	9
2.1.17	getContextURI() . . . . .	10
2.1.18	login() . . . . .	10
2.1.19	parse(HttpResponse, Class<U>) . . . . .	10
2.1.20	stringified(Object) . . . . .	10

<b>3</b>	<b>Paket com.frentix.restapi.tool</b>	<b>11</b>
3.1	Klasse Course . . . . .	11
3.1.1	Syntax . . . . .	11
3.1.2	Course(RestConnection) . . . . .	11
3.1.3	importCourse(File, File) . . . . .	12
3.1.4	queryCourse(File) . . . . .	12
3.2	Klasse RepositoryEntry . . . . .	12
3.2.1	Syntax . . . . .	12
3.2.2	RepositoryEntry(RestConnection) . . . . .	13
3.2.3	importRepositoryEntry(File, File) . . . . .	13
3.2.4	queryRepositoryEntry(File) . . . . .	13
3.3	Klasse User . . . . .	13
3.3.1	Syntax . . . . .	13
3.3.2	User(RestConnection) . . . . .	15
3.3.3	User(RestConnection, String, String, String) . . . . .	15
3.3.4	applyProperties(Properties) . . . . .	15
3.3.5	generateUser(File, int) . . . . .	15
3.3.6	importUser(File, File) . . . . .	16
3.3.7	queryUser(File) . . . . .	16
<b>4</b>	<b>Paket com.frentix.util</b>	<b>17</b>
4.1	Klasse CsvHelper . . . . .	17
4.1.1	Syntax . . . . .	17
4.1.2	readCsv(File, char) . . . . .	18
4.1.3	writeCourseToCsv(File, char, boolean, CourseVO) . . . . .	18
4.1.4	writeCourseToCsv(OutputStream, char, CourseVO) . . . . .	19
4.1.5	writeUserToCsv(File, char, boolean, List<UserVO>) . . . . .	19
4.1.6	writeUserToCsv(File, char, boolean, UserVO) . . . . .	19
4.1.7	writeUserToCsv(OutputStream, char, UserVO) . . . . .	20
<b>A</b>	<b>Attribute mit konstanten Werten</b>	<b>21</b>
A.1	Paket com.frentix.* . . . . .	21
A.2	Paket com.frentix.restapi.* . . . . .	21
A.3	Paket com.frentix.restapi.tool.* . . . . .	21
<b>B</b>	<b>Statistiken</b>	<b>22</b>
B.1	Gesamtsummen . . . . .	22
B.2	Klassen pro Package . . . . .	23
<b>5</b>	<b>Index</b>	<b>24</b>

# Tabellenverzeichnis

A.1	Resttool . . . . .	21
A.2	RestConnection . . . . .	21
A.3	User . . . . .	21

## **Zusammenfassung**

# Kapitel 1

## Paket com.frentix

### 1.1 Klasse Resttool

You may import or query repository entries with Resttool as well generate, create or query users of an openolat LMS instance.

#### 1.1.1 Syntax

```
public
class
Resttool
{
// Public Static Fields
    public
    static
    final
    String
    CREATE_USER_USAGE
    = "-i user.csv -o user.csv"
;

    public
    static
    final
    String
    GENERATE_USER_USAGE
    = "-g user count -o user.csv"
;

    public
    static
    final
    String
    IMPORT_REPOSITORY_ENTRY_USAGE
    = "-i repositoryEntry.zip -o repositoryEntry.csv"
;

    public
    static
    final
    String
```

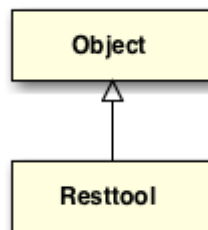
```
        QUERY_REPOSITORY_ENTRY_USAGE
        = "-q repository -o repositoryEntry.csv"
    ;

    public
    static
    final
    String
    QUERY_USER_USAGE
    = "-q user -o user.csv"
    ;

    public
    static
    final
    String
    RESTTOOL_INVOKE
    = "java -jar Resttool.jar"
    ;
// Public Constructors  public Resttool();
// Public Static Methods  public static void main(String[] args)
//                        throws NumberFormatException, IOException, URISyntaxException;
    public static void printUsage();
}
```

**Methoden geerbt von java.lang.Object:** clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait

**Autor** jkraehemann, joel.kraehemann@frentix.com, <http://www.frentix.com>



### 1.1.2 printUsage()

```
public static void printUsage();
```

Shows sample usage on console.



## Kapitel 2

# Paket com.frentix.restapi

### 2.1 Klasse RestConnection

Description: Manage a connection to the grizzly server used by the unit test with some helpers methods. Initial Date: 20 d√/©c. 2011

#### 2.1.1 Syntax

```
public
class
RestConnection
{
// Public Static Fields
public
static
final
String
CONTEXT_PATH
= "openolat"
;

public
static
final
String
HOST
= "localhost"
;

public
static
final
String
LOGIN
= "administrator"
;

public
static
final
```

```
        String
        PASSWORD
        = "openolat"
    ;

    public
    static
    final
    int
    PORT
    = 9998
    ;

    public
    static
    final
    String
    PROTOCOL
    = "http"
    ;

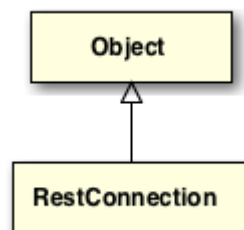
// Public Constructors  public RestConnection();
    public RestConnection(String host,
                           int port,
                           String protocol,
                           String contextPath,
                           String login,
                           String password);
    public RestConnection(URL url);
// Public Methods  public void addEntity(HttpEntityEnclosingRequestBase put,
        NameValuePair[] pairs)
        throws UnsupportedEncodingException;
    public void addJsonEntity(HttpEntityEnclosingRequestBase put,
                              Object obj)
        throws UnsupportedEncodingException;
    public void addMultipart(HttpEntityEnclosingRequestBase post,
                              String filename,
                              File file)
        throws UnsupportedEncodingException;
    public void applyProperties(Properties properties);
    public HttpDelete createDelete(URI uri,
                                    String accept,
                                    boolean cookie);
    public HttpGet createGet(URI uri,
                              String accept,
                              boolean cookie);
    public HttpPost createPost(URI uri,
                                String accept,
                                boolean cookie);
    public HttpPut createPut(URI uri,
                              String accept,
                              boolean cookie);
    public HttpPut createPut(URI uri,
                              String accept,
                              String langage,
                              boolean cookie);
    public HttpResponse execute(HttpUriRequest request)
        throws IOException, URISyntaxException;
```

---

```
public T get (URI uri,
             Class<T> cl)
    throws IOException, URISyntaxException;
public UriBuilder getBaseURI ()
    throws URISyntaxException;
public String getContextPath ();
public UriBuilder getContextURI ()
    throws URISyntaxException;
public CookieStore getCookieStore ();
public String getHost ();
public String getLogin ();
public String getPassword ();
public int getPort ();
public String getProtocol ();
public String getSecurityToken ();
public String getSecurityToken (HttpResponse response);
public boolean login ()
    throws IOException, URISyntaxException;
public boolean login (String username,
                    String password)
    throws IOException, URISyntaxException;
public U parse (HttpResponse response,
              Class<U> cl);
public void setContextPath (String contextPath);
public void setCredentials (String username,
                          String password);
public void setHost (String host);
public void setLogin (String login);
public void setPassword (String password);
public void setPort (int port);
public void setProtocol (String protocol);
public void shutdown ();
public String stringuified (Object obj);
}
```

**Methoden geerbt von java.lang.Object:** clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait

**Autor** srosse, [stephane.rosse@frentix.com](mailto:stephane.rosse@frentix.com), <http://www.frentix.com>, jkraehemann, [joel.kraehemann@frentix.com](mailto:joel.kraehemann@frentix.com), <http://www.frentix.com>



### 2.1.2 RestConnection()

```
public RestConnection ();
```

Default constructor creates a new RestConnection instance.

### 2.1.3 RestConnection(String, int, String, String, String, String)

```
public RestConnection(String host,
                      int port,
                      String protocol,
                      String contextPath,
                      String login,
                      String password);
```

Creates a new RestConnection instance.

Parameter	
host	remote host
port	the port to connect
protocol	the protocol to be used
contextPath	the default context path
login	credentials
password	credentials

### 2.1.4 RestConnection(URL)

```
public RestConnection(URL url);
```

Creates a new RestConnection instance.

Parameter	
url	URL to remote location.

### 2.1.5 addEntity(HttpEntityEnclosingRequestBase, NameValuePair...)

```
public void addEntity(HttpEntityEnclosingRequestBase put,
                     NameValuePair[] pairs)
    throws UnsupportedOperationException;
```

Add an object

Parameter	
put	the PUT request
pairs	the names and corresponding values

#### EXCEPTIONS

**UnsupportedEncodingException** java.io.UnsupportedEncodingException

### 2.1.6 addJsonEntity(HttpEntityEnclosingRequestBase, Object)

```
public void addJsonEntity(HttpEntityEnclosingRequestBase put,
                          Object obj)
    throws UnsupportedOperationException;
```

Add an object (application/json)

---

---

Parameter	
put	the PUT request
obj	the JSON

#### EXCEPTIONS

**UnsupportedEncodingException** java.io.UnsupportedEncodingException

### 2.1.7 addMultipart(HttpEntityEnclosingRequestBase, String, File)

```
public void addMultipart (HttpEntityEnclosingRequestBase post,
                        String filename,
                        File file)
    throws UnsupportedEncodingException;
```

Attach file to POST request.

Parameter	
post	the request
filename	the filename field
file	the file

#### EXCEPTIONS

**UnsupportedEncodingException** java.io.UnsupportedEncodingException

### 2.1.8 applyProperties(Properties)

```
public void applyProperties (Properties properties);
```

Applies the given properties.

Parameter	
properties	the Properties

### 2.1.9 createDelete(Uri, String, boolean)

```
public HttpDelete createDelete (Uri uri,
                                String accept,
                                boolean cookie);
```

HTTP DELETE method.

Parameter	
uri	
accept	
cookie	
<i>return</i>	

### 2.1.10 createGet(Uri, String, boolean)

```
public HttpGet createGet(Uri uri,  
                        String accept,  
                        boolean cookie);
```

HTTP GET method.

Parameter	
uri	
accept	
cookie	
return	

### 2.1.11 createPost(Uri, String, boolean)

```
public HttpPost createPost(Uri uri,  
                          String accept,  
                          boolean cookie);
```

HTTP POST method.

Parameter	
uri	
accept	
cookie	
return	

### 2.1.12 createPut(Uri, String, boolean)

```
public HttpPut createPut(Uri uri,  
                        String accept,  
                        boolean cookie);
```

HTTP PUT method.

Parameter	
uri	
accept	
cookie	
return	

### 2.1.13 createPut(Uri, String, String, boolean)

```
public HttpPut createPut(Uri uri,  
                        String accept,  
                        String language,  
                        boolean cookie);
```

HTTP PUT method.

Parameter	
uri	

<code>accept</code>	
<code>language</code>	
<code>cookie</code>	
<code>return</code>	

### 2.1.14 `execute(HttpUriRequest)`

```
public HttpResponse execute(HttpUriRequest request)
    throws IOException, URISyntaxException;
```

Execute request.

Parameter	
<code>request</code>	the request
<code>return</code>	the HttpResponse

#### EXCEPTIONS

**IOException** java.io.IOException

**URISyntaxException** java.net.URISyntaxException

### 2.1.15 `get(URI, Class<T>)`

```
public T get(URI uri,
            Class<T> cl)
    throws IOException, URISyntaxException;
```

HTTP GET method.

Parameter	
<code>uri</code>	the resource
<code>cl</code>	the vo to send
<code>return</code>	the retrieved vo

#### EXCEPTIONS

**IOException** java.io.IOException

**URISyntaxException** java.net.URISyntaxException

### 2.1.16 `getBaseURI()`

```
public UriBuilder getBaseURI()
    throws URISyntaxException;
```

Parameter	
<code>return</code>	http://localhost:9998

### 2.1.17 getContextURI()

```
public UriBuilder getContextURI()  
    throws URISyntaxException;
```

Parameter	
<i>return</i>	http://localhost:9998/olat

### 2.1.18 login()

```
public boolean login()  
    throws IOException, URISyntaxException;
```

Login to RESTful webservice.

Parameter	
<i>return</i>	

#### EXCEPTIONS

**IOException** java.io.IOException

**URISyntaxException** java.net.URISyntaxException

### 2.1.19 parse(HttpResponse, Class<U>)

```
public U parse(HttpResponse response,  
               Class<U> cl);
```

Parses the response.

Parameter	
response	the HttpResponse
cl	the VO's class
<i>return</i>	the parsed VO

### 2.1.20 stringuified(Object)

```
public String stringuified(Object obj);
```

Make string representation of a JSON object.

Parameter	
obj	
<i>return</i>	



## Kapitel 3

# Paket com.frentix.restapi.tool

### 3.1 Klasse Course

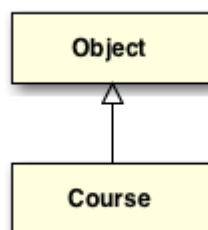
Import or query courses via RESTful web service.

#### 3.1.1 Syntax

```
public
class
Course
{
// Public Constructors  public Course(RestConnection restConnection);
// Public Methods  public RestConnection getRestConnection();
    public void importCourse(File outputFile,
                             File inputFile)
        throws IOException, URISyntaxException;
    public void queryCourse(File outputFile);
    public void setRestConnection(RestConnection restConnection);
}
```

**Methoden geerbt von java.lang.Object:** clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait

**Autor** jkraehemann, joel.kraehemann@frentix.com, <http://www.frentix.com>



#### 3.1.2 Course(RestConnection)

```
public Course(RestConnection restConnection);
```

Creates a new `Course` instance.

---

Parameter	
restConnection	the RestConnection

### 3.1.3 importCourse(File, File)

```
public void importCourse(File outputFile,
                        File inputFile)
    throws IOException, URISyntaxException;
```

Imports a course via RESTful web service.

Parameter	
outputFile	the file to save the course's attributes.
inputFile	the course as a ZIP file

#### EXCEPTIONS

**IOException** java.io.IOException

**URISyntaxException**

### 3.1.4 queryCourse(File)

```
public void queryCourse(File outputFile);
```

Queries courses via RESTful web service.

Parameter	
outputFile	the file to save the queried courses

## 3.2 Klasse RepositoryEntry

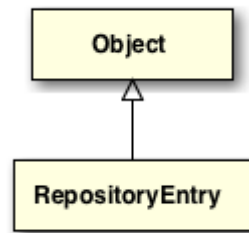
Import or query repository entries via RESTful web service.

### 3.2.1 Syntax

```
public
class
RepositoryEntry
{
// Public Constructors  public RepositoryEntry(RestConnection restConnection);
// Public Methods  public RestConnection getRestConnection();
    public void importRepositoryEntry(File outputFile,
                                    File inputFile);
    public void queryRepositoryEntry(File outputFile);
    public void setRestConnection(RestConnection restConnection);
}
```

**Methoden geerbt von java.lang.Object:** clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait

**Autor** jkraehemann, joel.kraehemann@frentix.com, <http://www.frentix.com>



### 3.2.2 RepositoryEntry(RestConnection)

```
public RepositoryEntry(RestConnection restConnection);
```

Creates a new RepositoryEntry instance.

Parameter	
restConnection	the RestConnection

### 3.2.3 importRepositoryEntry(File, File)

```
public void importRepositoryEntry(File outputFile,
                                   File inputFile);
```

Imports a repository entry via RESTful web service.

Parameter	
outputFile	the file to save the repository entry's attributes.
inputFile	the repository entry as a ZIP file

### 3.2.4 queryRepositoryEntry(File)

```
public void queryRepositoryEntry(File outputFile);
```

Queries repository entries via RESTful web service.

Parameter	
outputFile	the file to save the queried repository entries

## 3.3 Klasse User

Generate, import or query users via RESTful web service. You may set the appropriate properties to configure parameters of user generation.

### 3.3.1 Syntax

```
public
class
```

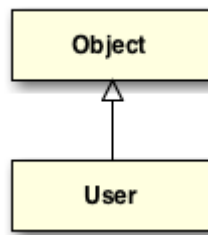
```
        User
    {
// Public Static Fields
        public
        static
        final
        String
        GENERATE_USER_FIRSTNAME_PREFIX
        = "John"
    ;

        public
        static
        final
        String
        GENERATE_USER_LOGIN_PREFIX
        = "jmeter_"
    ;

        public
        static
        final
        String
        GENERATE_USER_SURNAME_PREFIX
        = "JMeter_"
    ;
// Public Constructors    public User(RestConnection restConnection);
        public User(RestConnection restConnection,
                    String generateUserFirstnamePrefix,
                    String generateUserSurnamePrefix,
                    String generateUserLoginPrefix);
// Public Methods    public void applyProperties(Properties properties);
        public void generateUser(File outputFile,
                                int count)
            throws IOException, URISyntaxException;
        public String getGenerateUserFirstnamePrefix();
        public String getGenerateUserLoginPrefix();
        public String getGenerateUserSurnamePrefix();
        public RestConnection getRestConnection();
        public void importUser(File outputFile,
                                File inputFile);
        public void queryUser(File outputFile);
        public void setGenerateUserFirstnamePrefix(String generateUserFirstnamePrefix);
        public void setGenerateUserLoginPrefix(String generateUserLoginPrefix);
        public void setGenerateUserSurnamePrefix(String generateUserSurnamePrefix);
        public void setRestConnection(RestConnection restConnection);
    }
}
```

**Methoden geerbt von java.lang.Object:** clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait

**Autor** jkraehemann, joel.kraehemann@frentix.com, <http://www.frentix.com>



### 3.3.2 User(RestConnection)

```
public User(RestConnection restConnection);
```

Creates a new User instance.

Parameter	
restConnection	the RestConnection

### 3.3.3 User(RestConnection, String, String, String)

```
public User(RestConnection restConnection,
            String generateUserFirstnamePrefix,
            String generateUserSurnamePrefix,
            String generateUserLoginPrefix);
```

Creates a new User instance.

Parameter	
restConnection	the RestConnection
generateUserFirstnamePrefix	prefix to be used
generateUserSurnamePrefix	prefix to be used
generateUserLoginPrefix	prefix to be used

### 3.3.4 applyProperties(Properties)

```
public void applyProperties(Properties properties);
```

Applies the given properties.

Parameter	
properties	the Properties

### 3.3.5 generateUser(File, int)

```
public void generateUser(File outputFile,
                        int count)
    throws IOException, URISyntaxException;
```

Generates users via RESTful web service.

---

---

Parameter	
outputFile	the file to save the generated user attributes
count	the count of users to be created

## EXCEPTIONS

**URISyntaxException** java.net.URISyntaxException

**IOException** java.io.IOException

*Siehe auch* , ,

### 3.3.6 importUser(File, File)

```
public void importUser(File outputFile,
                      File inputFile);
```

Imports users via RESTful web service.

Parameter	
inputFile	the file containing the user's attributes.
outputFile	

### 3.3.7 queryUser(File)

```
public void queryUser(File outputFile);
```

Queries existing users via RESTful web service.

Parameter	
outputFile	the file to save the queried user attributes.

## Kapitel 4

# Paket com.frentix.util

## 4.1 Klasse CsvHelper

Some helper methods for handling comma separated values (CSV) files.

### 4.1.1 Syntax

```

        public
        class
        CsvHelper
    {
// Public Constructors    public CsvHelper();
// Public Static Methods    public static String[] readCsv(File csvFile,
                                char separator)

        throws IOException;
    public static void writeCourseToCsv(File csvFile,
                                char separator,
                                boolean append,
                                CourseVO course)

        throws IOException;
    public static void writeCourseToCsv(OutputStream output,
                                char separator,
                                CourseVO course)

        throws IOException;
    public static void writeUserToCsv(File csvFile,
                                char separator,
                                boolean append,
                                UserVO user)

        throws IOException;
    public static void writeUserToCsv(File csvFile,
                                char separator,
                                boolean append,
                                List<UserVO> user)

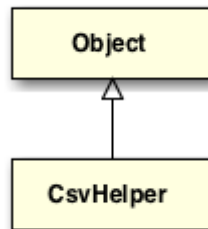
        throws IOException;
    public static void writeUserToCsv(OutputStream output,
                                char separator,
                                UserVO user)

        throws IOException;
    }

```

**Methoden geerbt von java.lang.Object:** clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait

**Autor** jkraehemann, joel.kraehemann@frentix.com, <http://www.frentix.com>



#### 4.1.2 readCsv(File, char)

```
public static String[] readCsv(File csvFile,
                               char separator)
    throws IOException;
```

Reads a CSV file into a string array.

Parameter	
csvFile	the file to be read
separator	the field separator
return	a string array whereas each field has its own entry

#### EXCEPTIONS

**IOException** java.io.IOException

#### 4.1.3 writeCourseToCsv(File, char, boolean, CourseVO)

```
public static void writeCourseToCsv(File csvFile,
                                     char separator,
                                     boolean append,
                                     CourseVO course)
    throws IOException;
```

Writes the course's attributes to CSV file you may specify whether to overwrite or to append the file.

Parameter	
csvFile	
separator	
append	
course	

#### EXCEPTIONS

**IOException** java.io.IOException

---



#### 4.1.4 writeCourseToCsv(OutputStream, char, CourseVO)

```
public static void writeCourseToCsv(OutputStream output,
                                   char separator,
                                   CourseVO course)
    throws IOException;
```

Writes the course's attributes to CSV output stream.

Parameter	
output	the OutputStream
separator	the field separator
course	the CourseVO

##### EXCEPTIONS

**IOException** java.io.IOException

#### 4.1.5 writeUserToCsv(File, char, boolean, List<UserVO>)

```
public static void writeUserToCsv(File csvFile,
                                   char separator,
                                   boolean append,
                                   List<UserVO> user)
    throws IOException;
```

Writes the attributes of the users to CSV file you may specify whether to overwrite or to append the file.

Parameter	
csvFile	the file to write
separator	the field separator
append	if false the file will be deleted first else the users will be append to end
user	the UserVOes

##### EXCEPTIONS

**IOException** java.io.IOException

#### 4.1.6 writeUserToCsv(File, char, boolean, UserVO)

```
public static void writeUserToCsv(File csvFile,
                                   char separator,
                                   boolean append,
                                   UserVO user)
    throws IOException;
```

Writes the user's attributes to CSV file you may specify whether to overwrite or to append the file.

Parameter	
csvFile	the file to write
separator	the field separator
append	if false the file will be deleted first else the user will be append to end
user	the UserVO

## EXCEPTIONS

**IOException** java.io.IOException**4.1.7 writeUserToCsv(OutputStream, char, UserVO)**

```
public static void writeUserToCsv(OutputStream output,  
                                char separator,  
                                UserVO user)  
    throws IOException;
```

Writes the user's attributes to CSV ouput stream.

Parameter	
output	the OuputStream
separator	the field separator
user	the UserVO

## EXCEPTIONS

**IOException** java.io.IOException

## Anhang A

# Attribute mit konstanten Werten

### A.1 Paket com.frentix.\*

CREATE_USER_USAGE	"-i user.csv -o user.csv"
GENERATE_USER_USAGE	"-g user count -o user.csv"
IMPORT_REPOSITORY_ENTRY_USAGE	"-i repositoryEntry.zip -o repositoryEntry.csv"
QUERY_REPOSITORY_ENTRY_USAGE	"-q repository -o repositoryEntry.csv"
QUERY_USER_USAGE	"-q user -o user.csv"
RESTTOOL_INVOKE	"java -jar Resttool.jar"

Tabelle A.1: Resttool

### A.2 Paket com.frentix.restapi.\*

CONTEXT_PATH	"openolat"
HOST	"localhost"
LOGIN	"administrator"
PASSWORD	"openolat"
PORT	9998
PROTOCOL	"http"

Tabelle A.2: RestConnection

### A.3 Paket com.frentix.restapi.tool.\*

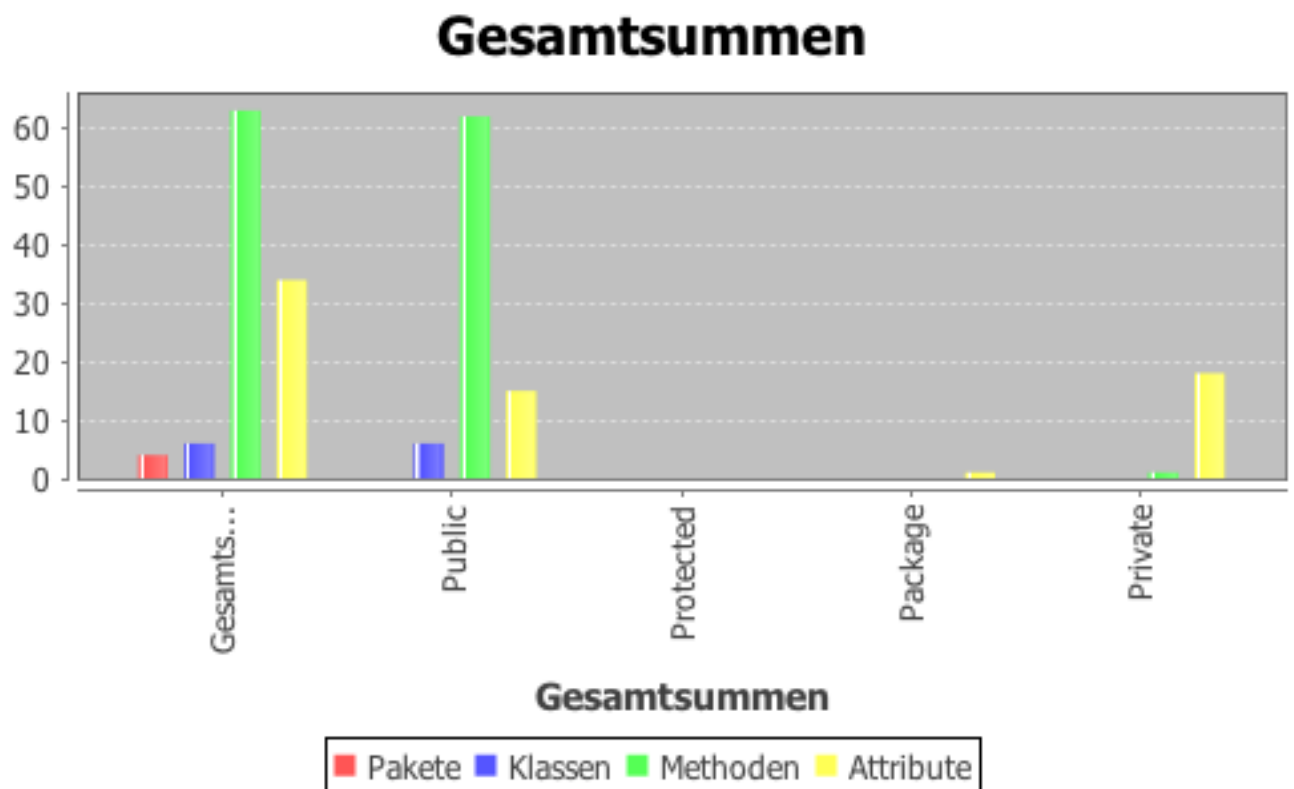
GENERATE_USER_FIRSTNAME_PREFIX	"John"
GENERATE_USER_LOGIN_PREFIX	"jmeter_"
GENERATE_USER_SURNAME_PREFIX	"JMeter_"

Tabelle A.3: User

## Anhang B

# Statistiken

### B.1 Gesamtsummen



Gesamtsummen	
Gesamtsummen Pakete	4
Gesamtsummen Klassen	6
Gesamtsummen Attribute	34
Gesamtsummen Methoden	63
Public Klassen	6
Public Attribute	15
Public Methoden	62
Protected Attribute	0
Protected Methoden	0
Package Klassen	0

Package Attribute	1
Package Methoden	0
Private Attribute	18
Private Methoden	1

B.2 Klassen pro Package



Klassen pro Package	
com.frentix.restapi.tool	3
com.frentix	1
com.frentix.restapi	1
com.frentix.util	1

# Kapitel 5

## Index

### A

addEntity, 6  
addJsonEntity, 6  
addMultipart, 7  
applyProperties, 7, 15

### C

#### Classes

Course, 11  
CsvHelper, 17  
RepositoryEntry, 12  
RestConnection, 3  
Resttool, 1  
User, 13

Course, 11  
createDelete, 7  
createGet, 8  
createPost, 8  
createPut, 8  
CsvHelper, 17

### E

execute, 9

### G

generateUser, 15  
get, 9  
getBaseUrl, 9  
getContextUri, 10

### I

importCourse, 12  
importRepositoryEntry, 13  
importUser, 16

### L

login, 10

### M

#### Methods

addEntity, 6  
addJsonEntity, 6  
addMultipart, 7  
applyProperties, 7, 15

Course, 11  
createDelete, 7  
createGet, 8  
createPost, 8  
createPut, 8  
execute, 9  
generateUser, 15  
get, 9  
getBaseUrl, 9  
getContextUri, 10  
importCourse, 12  
importRepositoryEntry, 13  
importUser, 16  
login, 10  
parse, 10  
printUsage, 2  
queryCourse, 12  
queryRepositoryEntry, 13  
queryUser, 16  
readCsv, 18  
RepositoryEntry, 13  
RestConnection, 5, 6  
stringified, 10  
User, 15  
writeCourseToCsv, 18, 19  
writeUserToCsv, 19, 20

### P

parse, 10  
printUsage, 2

### Q

queryCourse, 12  
queryRepositoryEntry, 13  
queryUser, 16

### R

readCsv, 18  
RepositoryEntry, 12, 13  
RestConnection, 3, 5, 6  
Resttool, 1

### S

stringified, 10

**U**

User, [13](#), [15](#)

**W**

writeCourseToCsv, [18](#), [19](#)

writeUserToCsv, [19](#), [20](#)

---