TestRail4J

Release latest

Apr 15, 2019

Contents

Java HTTP-client for the TestRail API.

A feign HTTP-clients are represented by the *TestRailClient* interface for each type of model annotation. Annotated jackson2 and gson models are generated using the jsonschema2pojo plugin from the json schemas.

Requirements

- Java 8+
- Open feign core 10.1+

CHAPTER 2

Usage

TestRail4J represented by two client modules with gson and Jackson2 models respectively. This is done for the convenience of the end user, so as not to use unnecessary dependencies in the project if one of the presented models is already used in the project.

Add the client to the pom.xml file. For example jackson2:

```
<dependency>
    <groupId>org.touchbit.testrail4j</groupId>
    <artifactId>jackson2-feign-client</artifactId>
    <version>${testrail4j.version}</version>
</dependency>
```

Build client using *TestRailClientBuilder* and call the necessary method

```
import org.touchbit.testrail4j.jackson2.gson.feign.TestRailClientBuilder;
import org.touchbit.testrail4j.core.BasicAuth;
import org.touchbit.testrail4j.jackson2.model.Case;

public class Example {
    public static void main(String[] a) {
        TestRailClient client = TestRailClientBuilder
                                .build("user", "pass", "http://localhost");

        Project project = client.addProject("name", "announcement", true, 3);
        Section section = new Section()
                .withName(UUID.randomUUID().toString())
                .withDescription(UUID.randomUUID().toString());
        Section section = client.addSection(section, project.getId());
        Case caze = new Case()
                .withTitle("test_20190101201312")
                .withPriorityId(CRITICAL.getId())
                .withSuiteId(section.getSuiteId())
                .withRefs("JIRA-123")
                .withTypeId(ACCEPTANCE.getId())
```

```
                .withTemplateId(TEST_CASE_TEXT.getId())
                .withEstimate("1m 45s")
                .withCustomPreconds("withCustomPreconds")
                .withCustomSteps("withCustomSteps")
                .withCustomExpected("withCustomExpected")
                .withCustomStepsSeparated(null);
        Case caze = client.addCase(caze, caze.getSectionId());
        System.out.println(caze.getId());
    }
}
```

Restrictions

3.1 *TestRailClient#addCaseField(TRCaseField)*

The returned object for the method of creating a new test case custom field is not available until the correction of the
defect

Table of Contents

4.1 Customisation

4.1.1 TestRailClient

If you have a need to expand calls to TestRail (new version has been released), then you can always create your own interface inherited from TestRailClient with your own methods of working with API. For example:

```
public interface CustomTestRailClient extends TestRailClient {

    @RequestLine(value = "GET /index.php%3F/api/v2/get_results/{foo}/{bar}")
    List<TRResult> getResultsWithFooBar(@Param("foo") Long foo, @Param("bar") Long␣
↪bar);
}
```

Next you need to build your specific TestRail client.

```
public class Example {
    public static void main(String[] a) {
        CustomTestRailClient client = TestRailClientBuilder
                .build("login", "pass", "http://tr.host", CustomTestRailClient.class);
        client.getResultsWithFooBar("foooooo", "baaaaaar");
    }
}
```

4.1.2 TestRailClientBuilder

TestRailClientBuilder - not necessary to use. You can always build your own *TestRailClient* using *Feign.Builder()*

```
public class Example {
    public static void main(String[] a) {
        CustomTestRailClient client = new Feign.Builder()
```

(continues on next page)

```
            .encoder(new GsonEncoder())
            .decoder(new GsonDecoder())
            .requestInterceptor(new CustomAuthInterceptor())
            .errorDecoder(new CustomTestRailErrorDecoder())
            .target(CustomTestRailClient.class, "http://custom.tr");
        client.getResultsWithFooBar("foooooo", "baaaaaar");
    }
}
```

4.1.3 Query Maps

All QueryMaps are empty interfaces to allow the end user to create and use their own filter with presets. Secondly to level surprises from TR when updating the version of the server.

All existing default filters are presented in the package *org.touchbit.testrail4j.core.query.filter*

```
public class Example {
    public static void main(String[] a) {
        TestRailClient client = TestRailClientBuilder
                                    .build("user", "pass", "http://localhost");
        List<TRProject> projects = client
                            .getProjects(new GetProjectsFilter().
→withIsCompleted(true);
    }
}
```

4.2 Modules

4.2.1 Common

- testrail4j-core - contains a common implementation for TestRail4J HTTP clients. Used in modules: jackson2-feign-client, gson-feign-client.

4.2.2 Clients

- jackson2-feign-client - *jackson2* annotated interface for the *feign* http-client

```
<dependency>
    <groupId>org.touchbit.testrail4j</groupId>
    <artifactId>jackson2-feign-client</artifactId>
    <version>${testrail4j.version}</version>
</dependency>
```

- gson-feign-client - *gson* annotated interface for the *feign* http-client

```
<dependency>
    <groupId>org.touchbit.testrail4j</groupId>
    <artifactId>gson-feign-client</artifactId>
    <version>${testrail4j.version}</version>
</dependency>
```

4.2.3 Models

- jackson2-api-model - *jackson2* annotated models. Used in modules: jackson2-feign-client.

```
<dependency>
    <groupId>org.touchbit.testrail4j</groupId>
    <artifactId>jackson2-api-model</artifactId>
    <version>${testrail4j.version}</version>
</dependency>
```

- gson-api-model - *gson* annotated models. Used in modules: gson-feign-client.

```
<dependency>
    <groupId>org.touchbit.testrail4j</groupId>
    <artifactId>gson-api-model</artifactId>
    <version>${testrail4j.version}</version>
</dependency>
```

4.3 Dependency tree

4.3.1 Jackson2

jackson2-feign-client

```
org.touchbit.testrail4j:jackson2-feign-client:jar:2.1.1
+- org.touchbit.testrail4j:testrail4j-core:jar:2.1.1:compile
|  \- org.slf4j:slf4j-api:jar:1.7.25:compile
+- org.touchbit.testrail4j:jackson2-api-model:jar:2.1.1:compile
|  +- com.fasterxml.jackson.core:jackson-annotations:jar:2.9.7:compile
|  \- org.apache.commons:commons-lang3:jar:3.8.1:compile
+- io.github.openfeign:feign-core:jar:10.1.0:compile
\- io.github.openfeign:feign-jackson:jar:10.1.0:compile
   \- com.fasterxml.jackson.core:jackson-databind:jar:2.9.6:compile
      \- com.fasterxml.jackson.core:jackson-core:jar:2.9.6:compile
```

jackson2-api-model

```
org.touchbit.testrail4j:jackson2-api-model:jar:2.1.1
+- com.fasterxml.jackson.core:jackson-annotations:jar:2.9.7:compile
\- org.apache.commons:commons-lang3:jar:3.8.1:compile
```

4.3.2 Gson

gson-feign-client

```
org.touchbit.testrail4j:gson-feign-client:jar:2.1.1
+- org.touchbit.testrail4j:testrail4j-core:jar:2.1.1:compile
|  \- org.slf4j:slf4j-api:jar:1.7.25:compile
+- org.touchbit.testrail4j:gson-api-model:jar:2.1.1:compile
```

```
|  +- com.google.code.gson:gson:jar:2.8.5:compile
|  \- org.apache.commons:commons-lang3:jar:3.8.1:compile
+- io.github.openfeign:feign-core:jar:10.1.0:compile
\- io.github.openfeign:feign-gson:jar:10.1.0:compile
```

gson-api-model

```
org.touchbit.testrail4j:gson-api-model:jar:2.1.1
+- com.google.code.gson:gson:jar:2.8.5:compile
\- org.apache.commons:commons-lang3:jar:3.8.1:compile
```

4.3.3 Core

testrail4j-core

```
org.touchbit.testrail4j:testrail4j-core:jar:2.1.1
+- io.github.openfeign:feign-core:jar:10.1.0:compile
\- org.slf4j:slf4j-api:jar:1.7.25:compile
```

4.4 MIT License

- Copyright (c) 2019 TouchBIT.

- Copyright (c) 2019 Oleg Shaburov.

- Copyright (c) 2018 Maria Vasilenko.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.