# Parkour Documentation

***Release***

**Evgeny Anatskiy**

**Jan 30, 2017**

# Contents

Parkour is ... short description.

CHAPTER 1

# Contents

## 1.1 Installation

test

## 1.2 Parkour API Documentation

test

### 1.2.1 Request API

API operations on requests.

request.views.**delete_request**(*request*)

**POST /request/delete_request/** Delete request with a given id and all its libraries and samples.

**Parameters** **request_id** – request id

**Returns** {'success': not error, 'error': error}

**Return type** JSON response

request.views.**generate_deep_sequencing_request**(*request*)

Generate Deep Sequencing Request form in PDF.

request.views.**get_all**(*request*)

**GET /request/get_all/** Get the list of all requests.

**Returns** list with requests

**Return type** JSON response

request.views.**get_libraries_and_samples**(*request*)

**GET /request/libraries_and_samples/?request_id={request_id}** Get the list of all libraries and samples for a given request.

**Parameters** **request_id** – request id

**Returns** libraries and samples in a given request

**Return type** JSON reponse

request.views.**save_request**(*request*)

**POST /request/save_request/** Add new or edit an existing request.

**Parameters**

- **mode** – add/edit - controles the mode: either add or edit a request
- **request_id** – request id if mode='edit'; otherwise, ''
- **libraries** – list of library ids
- **samples** – list of samples ids
- **description** – request description

**Returns** {'success': not error, 'error': error}

**Return type** JSON response

request.views.**upload_deep_sequencing_request**(*request*)

Upload Deep Sequencing request with PI's signature and change request status to 1.

### 1.2.2 Library API

API operations on libraries.

library.views.**delete_library**(*request*)

**POST /library/delete_library/** Delete library with a given id.

**Parameters** **record_id** – library id

**Returns** {'success': not error, 'error': error}

**Return type** JSON response

library.views.**get_all**(*request*)

**GET /library/get_all/** Get the list of all libraries and samples.

**Returns** list with libraries and samples

**Return type** JSON response

library.views.**get_files**(*request*)

library.views.**get_library_protocols**(*request*)

**GET /library/get_library_protocols/** Get the list of all library protocols.

**Returns** list with library protocols

**Return type** JSON response

library.views.**get_library_type**(*request*)

**GET /library/get_library_type/?library_protocol_id={library_protocol_id}/** Get the list of all library types for a given library protocol.

**Parameters** **library_protocol_id** – library protocol id

**Returns** list with library types

**Return type** JSON response

library.views.**save_library**(*request*)
Add a new library or update an existing one.

library.views.**upload_files**(*request*)

### 1.2.3 Sample API

API operations on samples.

sample.views.**delete_sample**(*request*)
Delete sample with a given id.

sample.views.**get_files**(*request*)

sample.views.**get_nucleic_acid_types**(*request*)
Get the list of all nucleic acid types.

sample.views.**get_sample_protocols**(*request*)
Get the list of all sample protocols.

sample.views.**save_sample**(*request*)
Add a new sample or update an existing one.

sample.views.**upload_files**(*request*)

### 1.2.4 Quality Check API

API operations on Quality Check.

quality_check.views.**update**(*request*)
Update library or sample field(s). If a library/sample passes the quality check, then set its status to 2. Otherwise, set the status to -1 and notify the user about the failure via email.

### 1.2.5 Index Generator API

API operations on pools.

index_generator.views.**generate_indices**(*request*)
Generate indices for given libraries and samples.

index_generator.views.**pooling_tree**(*request*)
Get libraries ready for pooling.

index_generator.views.**save_pool**(*request*)
Create a pool after generating indices, add libraries and "converted" samples to it, update the pool size, and create a Library Preparation object and a Pooling object for each added library/sample.

index_generator.views.**update_index_type**(*request*)
Update Index Type for a given sample.

index_generator.views.**update_read_length**(*request*)
Update Read Length for a given librray or sample.

### 1.2.6 Library Preparation API

API operations on Library Preparation.

library_preparation.views.**download_benchtop_protocol**(*request*)
Generate Benchtop Protocol as XLS file for selected samples.

library_preparation.views.**edit**(*request*)
Edit Library Preparation object.

library_preparation.views.**get_all**(*request*)
Get the list of all samples for Library Preparation.

library_preparation.views.**upload_benchtop_protocol**(*request*)
Upload a file and add it to all samples with a givel Library Protocol.

### 1.2.7 Pooling API

API operations on Pooling.

pooling.views.**download_pooling_template**(*request*)

pooling.views.**edit**(*request*)
Edit Pooling object.

pooling.views.**get_all**(*request*)
Get the list of all libraries.

pooling.views.**get_request**(*record*, *record_type*)
Get request for a current library/sample.

pooling.views.**upload_pooling_template**(*request*)
Upload a file and attach it to a given Pool.

### 1.2.8 Flowcell API

API operations on flowcells.

flowcell.views.**get_all**(*request*)
Get the list of all Flowcells.

flowcell.views.**pool_info**(*request*)
Get additional information for a given pool.

flowcell.views.**pool_list**(*request*)
Get the list of pools for loading flowcells.

flowcell.views.**save**(*request*)
Save a new flowcell.

flowcell.views.**sequencer_list**(*request*)
Get the list of all sequencers.

CHAPTER 2

# Indices and tables

- genindex
- modindex
- search

# Python Module Index

# Index