Starexec Web Interface

Table of Contents

1 Introduction

1.1 Using the URLs

Unless otherwise specified, all URLs documented in the functions below are suffixes of the following URL, which points to the secure resources of Starexec.

https://www.starexec.org/starexec/secure/

1.2 Logging In

To be able to access Starexec resources, you will need to log into Starexec and maintain a secure channel. Logging in is a multistep protocol, with 3 main steps. First, authorization works on a challenge-based system, which means you cannot log in until you request a secure resource. Next, you must provide your username and password, and finally, you will access a secure resource.

First Step – Requesting a Secure Resource.

URL: index.jsp

Method: GET

Parameter Encoding: N/A

Returns: The login page

Return Cookies

JSESSIONID : Integer – On success, contains a session ID that you will need to use in later steps.

Second Step – Providing Login Credentials

URL: j_security_check

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

j_username : String – Your username

j_password : String – Your password

cookieexists : Boolean – Should be set to false for web API calls.

Description: Logs you into Starexec. You must provide the JSESSIONID you acquired in the previous steps.

Returns: A jSON string containing a status object

Third Step – Accessing Secure Resources

URL: index.jsp

Method: GET

Parameter Encoding: N/A

Returns: The login page

Return Cookies

JSESSIONID : Integer – On success, contains a new session ID. You must begin using this new session ID for all subsequent interactions.

1.3 Maintaining a Session

After you have logged in, you must maintain a session for as long as you want to make requests to the secure resources of Starexec. To maintain a session, your HTTP requests should have the following header keys with the appropriate values.

Cookie : String – Should contain a string of the form "killmenothing; JSESSIONID=" + your JSESSIONID obtained when logging in.

Connection : String – Should be "keep-alive"

Accept-Language : String – Should be "en-US,en;q=0.5"

1.4 Permissions

Several functions below accept all of the following permissions parameters.

addBench : Boolean – Whether default permissions for this space should include adding benchmarks

addJob : Boolean – Whether default permissions for this space should include adding jobs

addSolver : Boolean – Whether default permissions for this space should include adding solvers

addSpace : Boolean – Whether default permissions for this space should include adding spaces

addUser : Boolean – Whether default permissions for this space should include adding users

removeBench : Boolean – Whether default permissions for this space should include removing benchmarks

removeJob : Boolean – Whether default permissions for this space should include removing jobs

removeSolver : Boolean – Whether default permissions for this space should include removing solvers

removeSpace : Boolean – Whether default permissions for this space should include removing spaces

removeUser : Boolean – Whether default permissions for this space should include removing users

isLeader : Boolean – Whether the a user should be a leader or not.

2 Spaces

2.1 Add Space

URL: add/space

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

parent : Integer – The ID of the parent space of this new space.

name : String – The new space name

desc : String – The new space description

locked : Boolean – Whether the space should be locked

users : Boolean – True to inherit all users from the parent space and false otherwise.

sticky : Boolean – Whether the new space should have sticky leaders set

+ all permissions parameters except isLeader (see the permissions section) : Specifies the default permissions for new users being added to the space.

Description: Creates a new space using the given attributes.

Returns: An HTTP message with 200 status on success, and an HTTP message with an error status on failure.

Return Cookies

New_ID : Integer – On success, contains the ID of the newly created space.

2.2 Upload Space XML

URL: upload/space

Method: POST

Parameter Encoding: multipart/form-data

Parameters

space : Integer – The ID of the space to use as the parent space for the new upload

f : File – The archive file containing the space XML.

Description: Uploads a space XML file and creates new spaces based on the XML

Returns: An HTTP redirect to the upload status page on success, and an HTTP message with an error status on failure.

2.3 Download Space

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be “space”

id : Integer – The ID of the space to download.

includesolvers : Boolean – Whether to include solvers in the download.

includebenchmarks : Boolean – Whether to include benchmarks in the download.

hierarchy : Boolean – True to download the space hierarchy rooted at the given space, and false to download only the given space.

Description: Makes a request to download an archive containing a directory representation of a space or space hierarchy, optionally including all the solvers and benchmarks in the spaces.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

2.4 Download Space XML

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be “spaceXML”

id : Integer – The ID of the space to get the XML for.

includeattrs : Boolean – Whether to include benchmark attributes in the XML.

updates : Boolean – Whether to include benchmark update tags in the XML.

upid : Integer – The ID of the update processor to include for all the update tags. Applies only if updates is true.

Description: Makes a request to download an archive containing the XML representation of a space hierarchy.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

2.5 Copy Spaces

URL: services/spaces/{spaceId}/copySpace

URL Variables

{spaceId} : Integer – The ID of the space that you want to copy other spaces into

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds : [Integer] – A list of space IDs, where each space will be copied into the destination space

copyHierarchy : Boolean – True to copy entire space hierarchies into the destination space. False to copy only the spaces in selectedIds without their hierarchies.

Description: Copies spaces into a single destination space. All benchmarks, solvers, and jobs will be linked into the newly created spaces.

Returns: A jSON string containing a status object.

2.6 Link all Orphaned Primitives to Space

URL: services/linkAllOrphaned/{userId}/{spaceId}

URL Variables

{userId} : Integer – Your user ID

{spaceId} : Integer– The ID of the space to place all of the orphaned primitives into.

Method: POST

Parameter Encoding: N/A

Description: Links all "orphaned" solvers, benchmarks, and jobs that you own in the given space.

Returns: A jSON string containing a status object.

2.7 Edit Space Attributes

URL: services/edit/space/{id}

URL Variables

{id} : Integer – The ID of the space to edit

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

name : String – The new space name

description : String – The new space description

locked : Boolean – Whether the space should be locked

sticky : Boolean – Whether the new space should have sticky leaders set

+ all permissions parameters (see the permissions section)

Description: Edits an existing space by providing a new values for all space attributes

Returns: A jSON string containing a status object.

2.8 Edit Single Space Attribute

URL: services/edit/space/{attr}/{id}

URL Variables

{id} : Integer – The ID of the space to edit

{attr} : String – The attribute to edit. Can be “name” or “description”

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

val : String -- The new value to use for the given attribute

Description: Edits an existing space by providing a new value for the given attribute

Returns: A jSON string containing a status object.

2.9 View Space Visibility

URL: services/space/isSpacePublic/{id}

URL Variables

{id} : Integer – The ID of the space to check visibility of.

Method: POST

Parameter Encoding: N/A

Description: Checks whether the given space is public or private.

Returns: A jSON string containing 1 if the space is public and 0 otherwise.

2.10 Edit Space Visibility

URL: services/space/changePublic/{id}/{hierarchy}/{makePublic}

URL Variables

{id} : Integer – The ID of the space to edit.

{hierarchy} : Boolean – True to edit the entire space hierarchy and false to edit the single space.

{makePublic}: Boolean – True to make spaces public and false to make them private

Method: POST

Parameter Encoding: N/A

Description: Sets the given space, and optionally the full space hierarchy rooted there, to either public or private.

Returns: A jSON string containing a status object.

2.11 Remove Spaces

URL: services/remove/subspace

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of space IDs to use

recyclePrims : Boolean – If true, all solvers and benchmarks located anywhere in any space hierarchy being removed will be recycled. Only solvers and benchmarks that you have permission to edit will be recycled.

Description: Removes all the given spaces, including their full space hierarchies.

Returns: A jSON string containing a status object.

3 Solvers

3.1 Upload Solver Archive

URL: upload/solvers

Method: POST

Parameter Encoding: multipart/form-data

Parameters

space : Integer – The ID of the space to upload the solver to.

sn : String – The name to give the solver.

descMethod : String – Describes the method that the solver description is being uploaded. Can be “text”, “file”, or “upload”. If “text” is used, then the string “desc” parameter should provide the description. If “file” is used, then the file object parameter “d” should be a text file containing the description. If “upload”, then the description will be taken from the uploaded archive.

desc : String – The string description for the solver. Applies only if descMethod is “text”.

d : File – If descMethod is “file”, then this is the file object.

upMethod : String – Can be “local” or “URL”. If URL, means benchmarks are being uploaded from a URL. Otherwise, means a file is being uploaded directly.

f : File – If a file is being uploaded, this is the file object.

url : String – If the upload is from a URL, gives the URL pointing to the solver archive.

dlable : Boolean – True if the solver should be downloadable and false otherwise.

runTestJob : Boolean – Whether to immediately start a new job using this solver as a test. Defaults to false.

setting : Integer – The ID of a setting profile to use for the test job creation. Applies only if runTestJob is true.

type : Integer – The type to give the new solver. See the help page documentation on solvers to see the solver type codes.

Description: Uploads a new solver. Also optionally creates a test job for the solver, which is a job with one job pair per configuration in the new solver. Test jobs are created using parameters from an existing settings profile.

Returns: An HTTP redirect to the spaces page on success, and an HTTP message with an error status on failure.

Return Cookies

New_ID : Integer – On success, contains the upload ID for this archive upload.

3.2 Upload Configuration

URL: upload/jobXML

Method: POST

Parameter Encoding: multipart/form-data

Parameters

solverId : Integer – The ID of the solver to attach this configuration to.

file : File – The file containing the configuration.

uploadConfigName : String – The name to give to the configuration.

uploadConfigDesc – The description to give to the configuration.

Description: Uploads a new configuration file to an existing solver.

Returns: An HTTP redirect to the solver details page on success, and an HTTP message with an error status on failure.

Return Cookies

New_ID : Integer – On success, contains the ID of the new configuration.

STATUS_MESSAGE_STRING: String – On failure, contains an error message.

3.3 Add Configuration

URL: save/configurations

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

solverId : Integer – The ID of the solver to attach this configuration to.

saveConfigContents : String – The new configuration as a string.

saveConfigName : String – The name to give to the configuration.

saveConfigDesc – The description to give to the configuration.

Description: Adds a new configuration to an existing solver.

Returns: An HTTP redirect to the solver details page on success, and an HTTP message with an error status on failure.

Return Cookies

New_ID : Integer – On success, contains the ID of the new configuration.

STATUS_MESSAGE_STRING: String – On failure, contains an error message.

3.4 Download Solver

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be "solver"

id : Integer – The ID of the solver to download.

reupload : Boolean – If true, solver will be downloaded in the format required for an upload to Starexec. Defaults to false.

Description: Makes a request to download an archive containing the given solver.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

3.5 Get Build Standard Output

URL: services/solvers/{id}/buildoutput

URL Variables

{id} : Integer – The ID of the solver to get the build output for

Method: GET

Parameter Encoding: N/A

Description: Retrieves the standard output of the build script used when the solver was uploaded

Returns: A plaintext string containing the standard output of the build script. Returns "not available" if no build script was used for this solver, if the output could not be found, or if you do not have permission to view the build output.

3.6 Linking / Copying Solvers to a Space

URL: services/spaces/{spaceId}/add/solver

URL Variables

{spaceId} : Integer – The ID of the space to put the solvers in

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of solver IDs to use

copy : Boolean – If true, deep copies of all the given solvers are made first, and then the new solvers are referenced in the given space. If false, solvers are simply referenced in the new space without being copied.

copyToSubspaces : Boolean – If true, solvers will be associated with every space in the hierarchy rooted at the given space. Otherwise, they will be associated only with the given space.

fromSpace : integer – If not null, then this is the ID of a space containing all the solvers in selectedIds[] that you have permission to copy solvers out of. If null, so such space is used, and you must be the owner of the solvers to have permission to use them.

Description: Given a list of solvers, places the benchmarks into the given space. If copy is true, the benchmarks are first copied. Otherwise, the benchmarks are just linked into the new space.

Returns: A jSON string containing a status object.

3.7 Remove Solvers From Space

URL: services/remove/solver/{spaceId}

URL Variables

{spaceId} : Integer – The ID of the space to remove solvers from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of solver IDs to use

hierarchy : Boolean – If true, removes the given solvers from the entire space hierarchy. Otherwise, just removes them from the given space.

Description: Removes all of the given solvers from the given space or space hierarchy, depending on the value of the hierarchy parameter. Note that the solvers are not recycled or deleted by this function

Returns: A jSON string containing a status object.

3.8 Simultaneously Recycle and Remove Solvers From a Space

URL: services/recycleandremove/solver/{spaceID}

URL Variables

{spaceId} : Integer – The ID of the space to remove solvers from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of solver IDs to use

Description: Simultaneously recycles all of the given solvers and removes them from the given space.

Returns: A jSON string containing a status object.

3.9 Edit Solver

URL: services/edit/solver/{id}

URL Variables

{id} : Integer – The ID of the solver to edit

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

name : String – The new name to give the solver

description : String – The new description to give the solver

downloadable : Boolean – Whether other users should be able to download the solver

Description: Modifies the given solver, giving it a new name, description, and downloadable status.

Returns: A jSON string containing a status object.

3.10 Edit Configuration

URL: services/edit/configuration/{id}

URL Variables

{id} : Integer – The ID of the configuration to edit.

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

name : String – The new name to give the configuration

description : String – The new description to give the configuration

Description: Modifies the given configuration, giving it a new name and description.

Returns: A jSON string containing a status object.

3.11 Recycle Solvers

URL: services/recycle/solver

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of solver IDs to use

Description: Recycles all of the given solvers. Note that the solvers will not be removed from any spaces they are currently associated with.

Returns: A jSON string containing a status object.

3.12 Recycle Orphaned Solvers

URL: services/recycleOrphaned/solver/{userId}

URL Variables

{userId} : Integer – Your user ID

Method: POST

Parameter Encoding: N/A

Description: Recycles all “orphaned” solvers that you own

Returns: A jSON string containing a status object.

3.13 Restore Solvers

URL: services/restore/solver

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of solver IDs to use

Description: Restores all of the given solvers, removing them from the recycle bin.

Returns: A jSON string containing a status object.

3.14 Restore Recycled Solvers

URL: services/restorerecycled/solvers

Method: POST

Parameter Encoding: N/A

Description: Restores all the solvers in your recycle bin, removing them from the recycle bin.

Returns: A jSON string containing a status object.

3.15 Delete Solvers

URL: services/delete/solver

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of solver IDs to use

Description: Permanently deletes all of the given solvers on disk. Note that the solvers will not be removed from any spaces they are currently associated with.

Returns: A jSON string containing a status object.

3.16 Delete Recycled Solvers

URL: services/deleterecycled/solvers

Method: POST

Parameter Encoding: N/A

Description: Permanently deletes all the solvers in your recycle bin.

Returns: A jSON string containing a status object.

3.17 Delete Configurations

URL: services/delete/configuration

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of configurations IDs to use

Description: Permanently deletes all of the given configurations on disk and removes them from their associated solvers.

Returns: A jSON string containing a status object.

4 Processors

4.1 Add Processor

URL: processors/manager

Method: POST

Parameter Encoding: multipart/form-data

Parameters

name : String – The new name to give the processor

desc : String – The new description to give the processor

file : File – The archive containing the processor

com : Integer – The ID of the community to put the solver in.

action : String – Should be “add” for a new processor

type : String – Describes the type of the processor. Should be “bench”, “pre”, “post”, or “update”.

Description: Modifies the given processor, giving it a new name and description.

Returns: A jSON string containing a status object.

4.2 Edit Processor

URL: services/edit/processor/{procId}

URL Variables

{procId} : Integer – The ID of the processor to edit

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

name : String – The new name to give the processor

desc : String – The new description to give the processor

Description: Modifies the given processor, giving it a new name and description.

Returns: A jSON string containing a status object.

4.3 Download Processors of Class

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be “proc”

id : Integer – The ID of the community that owns the processors to download.

procClass : Integer – The type of processor to download. Can be “post”, “pre”, or “bench” to download post processors, pre processors, and benchmark processors respectively.

Description: Makes a request to download an archive containing every benchmark of the given type that the given community owns.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

4.4 Delete Processors

URL: services/delete/processor

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of processor IDs to delete

Description: Permanently deletes the given processors

Returns: a jSON string containing a status object.

5 Benchmarks

5.1 Upload Benchmark Archive

URL: upload/benchmarks

Method: POST

Parameter Encoding: multipart/form-data

Parameters

space : Integer – The ID of the space to upload benchmarks to.

localOrUrl : String – Can be “local” or “URL”. If URL, means benchmarks are being uploaded from a URL. Otherwise, means a file is being uploaded directly.

benchFile : File – If a benchmark file is being uploaded, this is the file object.

url : String – If the upload is from a URL, gives the URL pointing to the benchmark archive.

upMethod : String – Can be either “convert” or “dump.” “Convert” means create a space hierarchy that mirrors the directory structure of the uploaded benchmarks, and “dump” means to simply dump all the benchmarks directly into the given space.

benchType : Integer – The ID of the benchmark processor that should be used to process the benchmarks.

download : Boolean – True if the benchmarks should be downloadable and false otherwise.

dependency : Boolean – True if these benchmarks have dependencies, and false otherwise.

depRoot : Integer – The ID of the space in which the benchmark dependencies are rooted, assuming dependency is true. Otherwise not necessary.

linked : Boolean – If dependency is true, then true if the first directory in path corresponds to dependent bench space.

+ all permissions attributes (see the permissions section). These will be the default permissions for all spaces created as a result of this upload.

Description: Uploads a new archive of benchmarks.

Returns: An HTTP redirect to the upload status page on success, and an HTTP message with an error status on failure.

Return Cookies

New_ID : Integer – On success, contains the upload ID for this archive upload.

STATUS_MESSAGE_STRING : String – On error, this is the error message string.

5.2 Getting Benchmark Upload Status

URL: services/benchmarks/uploadDescription/{statusId}

URL Variables

{statusId} : Integer – The ID of the benchmark upload status object

Method: GET

Parameter Encoding: N/A

Description: Retrieves a string description of a benchmark archive upload.

Returns: A jSON string containing a status object. If the status is successful, the message contained in the status object will be the upload description.

5.3 Download Benchmark

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be “bench”

id : Integer – The ID of the benchmark to download.

Description: Makes a request to download an archive containing the given benchmark.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

5.4 View Benchmark Contents

URL: services/benchmarks/{id}/contents

URL Variables

{id} : Integer – The ID of the benchmark to get the contents of

Method: GET

Parameter Encoding: N/A

Description: Retrieves the file contents of a single benchmark.

Returns: A plaintext string containing the benchmark contents for the given benchmark. Returns "not available" if the benchmark could not be found or if you do not have permission to see the given benchmark

5.5 Linking / Copying Benchmarks to a Space

URL: services/spaces/{spaceId}/add/benchmark

URL Variables

{spaceId} : Integer – The ID of the space to put the new benchmarks in

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of benchmark IDs to use

copy : Boolean – If true, benchmarks are first copied, and then the new benchmarks are referenced in the given space. If false, benchmarks are simply referenced in the new space without being copied.

fromSpace : integer – If not null, then this is the ID of a space containing all the benchmarks in selectedIds[] that you have permission to copy benchmarks out of. If null, so such space is used, and you must be the owner of the benchmarks to have permission to use them.

Description: Given a list of benchmarks, places the benchmarks into the given space. If copy is true, the benchmarks are first copied. Otherwise, the benchmarks are just linked into the new space.

Returns: A jSON string containing a status object.

5.6 Removing Benchmarks From a Space

URL: services/remove/benchmark/{spaceId}

URL Variables

{spaceId} : Integer – The ID of the space to remove benchmarks from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of benchmark IDs to use

Description: Given a list of benchmarks and a space, removes all the given benchmarks from the given space. Note that this does not either recycle or delete any of the given benchmarks.

Returns: A jSON string containing a status object.

5.7 Simultaneously Recycle and Remove Benchmarks From a Space

URL: services/recycleandremove/benchmark/{spaceID}

URL Variables

{spaceId} : Integer – The ID of the space to remove benchmarks from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of benchmark IDs to use

Description: Simultaneously recycles all of the given benchmarks and removes them from the given space.

Returns: A jSON string containing a status object.

5.8 Edit Benchmark

URL: services/edit/benchmark/{id}

URL Variables

{id} : Integer – The ID of the benchmark to edit

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

name : String – The new name to give the benchmark.

description : String – The new description to give the benchmark.

downloadable : Boolean – Whether other users should be able to download the benchmark

type : Integer – The ID of the benchmark processor to apply to the benchmark.

Description: Modifies the given benchmark, assigning it all the given attributes.

Returns: A jSON string containing a status object.

5.9 Reprocess Benchmarks in Space

URL: process/benchmarks

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

pid : Integer – The ID of the benchmark processor to use

sid : Integer – The ID of the space to reprocess benchmarks in.

hier : Boolean – True to reprocess all the benchmarks in the hierarchy rooted at the given space, and false to do only the single given space.

clear : Boolean – True to delete all old benchmark attributes, and false to only delete old attributes when there is a key conflict with the new attributes being created.

Description: Runs a new benchmark processor on all benchmarks in the given space (or hierarchy). Benchmark types are changed to the new processor, and all new benchmark attributes are saved. Old attributes may be optionally deleted; however, if there are any name conflicts between old and new attributes (for example, like having two “starexec-expected-results” attributes), then the old version will be deleted in every case.

Returns: An HTTP redirect to the upload status page on success, and an HTTP message with an error status on failure.

5.10 Recycle Benchmarks

URL: services/recycle/benchmark

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of benchmark IDs to use

Description: Recycles all of the given benchmarks. Note that the benchmarks will not be removed from any spaces they are currently associated with.

Returns: A jSON string containing a status object.

5.11 Recycle Orphaned Benchmarks

URL: services/recycleOrphaned/benchmark/{userId}

URL Variables

{userId} : Integer – Your user ID

Method: POST

Parameter Encoding: N/A

Description: Recycles all "orphaned" benchmarks that you own.

Returns: A jSON string containing a status object.

5.12 Restore Benchmarks

URL: services/restore/benchmark

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of benchmark IDs to use

Description: Restores all of the given benchmarks, removing them from the recycle bin.

Returns: A jSON string containing a status object.

5.13 Restore All Recycled Benchmarks

URL: services/restorerecycled/benchmarks

Method: POST

Parameter Encoding: N/A

Description: Restores all the benchmarks in your recycle bin, removing them from the recycle bin.

Returns: A jSON string containing a status object.

5.14 Delete Benchmarks

URL: services/delete/benchmark

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of benchmark IDs to use

Description: Permanently deletes all of the given benchmarks on disk. Note that the benchmarks will not be removed from any spaces they are currently associated with.

Returns: A jSON string containing a status object.

5.15 Delete Recycled Benchmarks

URL: services/deleterecycled/benchmarks

Method: POST

Parameter Encoding: N/A

Description: Permanently deletes all the benchmarks in your recycle bin.

Returns: A jSON string containing a status object.

6 Jobs

6.1 Create Job

URL: add/job

Method: POST

Parameter Encoding: multipart/form-data

Parameters

name : String – The name to give the job

desc : String – The description to give the job.

preProcess : Integer – The ID of the pre processor to use. Can be excluded.

seed : Integer – A number that will be passed into the pre processor for every pair.

postProcess : Integer – The ID of the post processor to use. Can be excluded.

queue : Integer – The ID of the queue to run the job on.

spaceId : Integer – The ID of the space to put the job in.

cpuTimeout : Integer – The CPU timeout, in seconds, to enforce.

wallclockTimeout : Integer – The wallclock timeout, in seconds, to enforce.

maxMem : Float – The maximum memory limit, in gigabytes.

pause : Boolean – If true, job will start out paused. If false, job will start upon creation.

runChoice : String – Controls how job pairs are created, and can be either “keepHierarchy” or “choose”. In “keepHierarchy”, a job is run using all benchmarks that are in the space hierarchy rooted at the spot that the job was created, and every benchmark is executed by every solver configuration of every solver in the same space. In “quickJob,” a single job pair is created, using the given solver and the given text to use as a new benchmark. In “choose”, a list of configurations is provided to use in the job.

configs : [Integer] – The list of configurations to use in the job. Only applies if runChoice is “choose”

benchChoice : String – Only applies if runChoice is “choose”. Describes how to select benchmarks for the job. Must be one of “runAllBenchInSpace”, “runAllBenchInHierarchy”, “runChosenFromSpace”. If “runAllBenchInSpace”, all benchmarks in the space the job is being uploaded to will be used. If

"runAllBenchInHierarchy", all benchmarks in the entire hierarchy will be used. If "runChosenFromSpace", then benchmarks must be provided.

bench : [Integer] – The list of benchmarks to use in the job. Only applies if benchChoice is "runChosenFromSpace".

traversal : String – Controls the order in which job pairs are executed. Can be either “depth” or “robin.” With “depth,” all the job pairs in a single space will be executed before moving onto another space. With “robin,” each space in the job will have a single pair executed before any space has a second pair executed, and so on.

Description: Creates a new job with the given parameters.

Returns: An HTTP redirect to the spaces page on success, and an HTTP message with an error code and error message on failure.

Return Cookies

New_ID : Integer – On success, contains the ID of the new job.

6.2 Create Quick Job

URL: add/job

Method: POST

Parameter Encoding: multipart/form-data

Parameters

name : String – The name to give the job

desc : String – The description to give the job.

preProcess : Integer – The ID of the pre processor to use. Can be excluded.

seed : Integer – A number that will be passed into the pre processor for every pair.

postProcess : Integer – The ID of the post processor to use. Can be excluded.

queue : Integer – The ID of the queue to run the job on.

spaceId : Integer – The ID of the space to put the job in.

cpuTimeout : Integer – The CPU timeout, in seconds, to enforce.

wallclockTimeout : Integer – The wallclock timeout, in seconds, to enforce.

maxMem : Float – The maximum memory limit, in gigabytes.

pause : Boolean – If true, job will start out paused. If false, job will start upon creation.

runChoice : String – Should be “quickJob”

bench : String – The benchmark to use as a string.

benchProcess : Integer – The ID of the benchmark processor to process the new benchmark with.

benchName : String – The name to give the new benchmark.

solver : Integer : The ID of the solver to use.

Description: Creates a new quick job, which is a job that uses every configuration of a single solver on a new benchmark, which is provided as a string.

Returns: An HTTP redirect to the upload status page on success, and an HTTP message with an error status on failure.

Return Cookies

New_ID : Integer – On success, contains the ID of the new job.

6.3 Upload Job XML

URL: upload/jobXML

Method: POST

Parameter Encoding: multipart/form-data

Parameters

space : Integer – The ID of the space to place the new jobs into.

f : File – The archive file containing the job XML.

Description: Uploads a job XML file and creates new jobs based on the XML.

Returns: An HTTP redirect to the space explorer on success, and an HTTP message with an error status on failure.

Return Cookies

STATUS_MESSAGE_STRING: String – On failure, contains an error message.

6.4 Linking Jobs in a New Space

URL: services/spaces/{spaceId}/add/job

URL Variables

{spaceId} : Integer – The ID of the space to put the jobs in

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of job IDs to use

fromSpace : integer – If not null, then this is the ID of a space containing all the jobs in selectedIds[] that you have permission to copy jobs out of. If null, so such space is used, and you must be the owner of the jobs to have permission to link them.

Description: Given a list of jobs, associates all the jobs with the given space.

Returns: A jSON string containing a status object.

6.5 Remove Jobs From Space

URL: services/remove/job/{spaceId}

URL Variables

{spaceId} : Integer – The ID of the space to remove jobs from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of job IDs to use

Description: Removes all of the given jobs from the given space.

Returns: A jSON string containing a status object.

6.6 Simultaneously Delete and Remove Jobs From a Space

URL: services/deleteandremove/job/{spaceID}

URL Variables

{spaceId} : Integer – The ID of the space to remove jobs from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of job IDs to use

Description: Simultaneously deletes all of the given jobs and removes them from the given space.

Returns: A jSON string containing a status object.

6.7 Download Job Output

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be "j_outputs"

id : Integer – The ID of the job.

since : Integer – A "completion number" that says to retrieve only job pairs with a completion number greater than the given one. Can be excluded to retrieve all pairs.

Description: Makes a request to download an archive containing the output files of all job pairs in this job, possibly with pairs that occurred before "since" excluded.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

6.8 Download Job CSV

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be "job"

id : Integer – The ID of the job.

since : Integer – A "completion number" that says to retrieve only job pairs with a completion number greater than the given one. Can be excluded to retrieve all pairs.

returnids : Boolean – If true, the CSV will include solver, configuration, and benchmark IDs.

getcompleted : Boolean – If true, only completed job pairs will be included. Defaults to false.

Description: Makes a request to download an archive containing the CSV representation of the given job.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

6.9 Download Job XML

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be "jobXML"

id : Integer – The ID of the job.

Description: Makes a request to download an archive containing the XML representation of the given job.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

6.10 Rerun Job Pairs

URL: services/jobs/rerunallpairs/{id}

URL Variables

{id} : Integer – The ID of the job that contains the pairs to be rerun

Method: POST

Parameter Encoding: N/A

Description: Reruns every job pair in the given job.

Returns: A jSON string containing a status object.

6.11 Rerun Job Pairs of a Status

URL: services/jobs/rerunpairs/{id}/{status}

URL Variables

{id} : Integer – The ID of the job that contains the pairs to be rerun

{status} : Integer – The integer status code of the pairs to rerun

Method: POST

Parameter Encoding: N/A

Description: Reruns all job pairs that are both in the given job and have the given status code.

Returns: A jSON string containing a status object.

6.12 Rerun Job Pairs That Reported Incorrectly

URL: services/jobs/rerunpairs/{id}

URL Variables

{id} : Integer – The ID of the job that contains the pairs to be rerun

Method: POST

Parameter Encoding: N/A

Description: Reruns all job pairs that have already completed, but have a CPU time or wallclock time of 0. Such pairs generally had some problem reporting back results after completing.

Returns: A jSON string containing a status object.

6.13 Reprocess Job Pairs

URL: services/postprocess/job/{jobId}/{procId}/{stageNumber}

URL Variables

{jobId} : Integer – The ID of the job to reprocess pairs in

{procId} : Integer – The ID of the post processor to use

{stageNumber} : Integer – Which stage of the job to reprocess. If the job did not use solver pipelines, this should always be 1.

Method: POST

Parameter Encoding: N/A

Description: Runs a new post processor on all pairs from the job. The job must be finished to run a new post processor.

Returns: A jSON string containing a status object.

6.14 Pause Job

URL: services/pause/job/{id}

URL Variables

{id} : Integer – The ID of the job to pause.

Method: POST

Parameter Encoding: N/A

Description: Pauses the given job.

Returns: A jSON string containing a status object.

6.15 Resume Job

URL: services/resume/job/{id}

URL Variables

{id} : Integer – The ID of the job to resume.

Method: POST

Parameter Encoding: N/A

Description: Resumes the given paused job.

Returns: A jSON string containing a status object.

6.16 Change Job Queue

URL: services/changeQueue/job/{id}/{queueid}

URL Variables

{id} : Integer – The ID of the job that is changing queues

{queueid} : Integer – The ID of the new queue to use

Method: POST

Parameter Encoding: N/A

Description: Moves the job to a new queue, so that all subsequent job pairs will run on the new queue.

Returns: A jSON string containing a status object.

6.17 View Qstat Output

URL: services/cluster/qstat

Method: GET

Parameter Encoding: N/A

Description: Executes qstat –f on the Starexec grid engine and gets the standard output. Used to determine the cluster status.

Returns: A plaintext string containing the results of qstat.

6.18 Delete Jobs

URL: services/delete/job

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of job IDs to use

Description: Permanently deletes all of the given jobs, including the deletion of all job data on disk.

Returns: A jSON string containing a status object.

7 Job Pairs

7.1 Download Single Job Pair Output

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be “jp_output”

id : Integer – The ID of the job pair.

Description: Makes a request to download an archive containing the output of the given job pair.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

7.2 Download Multiple Job Pair Outputs

URL: download

Method: GET

Parameter Encoding: URL Encoded

Parameters

type : String – Should be "jp_outputs"

id[] : [Integer] – The IDs of all the job pairs to get output of.

Description: Makes a request to download an archive containing the output files for all of the given job pairs.

Returns: An HTTP response with an output stream for the request archive on success. On failure, the response will contain an HTTP error code and a page with an error message.

7.3 Rerun Single Job Pair

URL: services/jobs/pairs/rerun/{pairid}

URL Variables

{pairid} : Integer – The ID of the job pair that needs to be rerun

Method: POST

Parameter Encoding: N/A

Description: Reruns the given job pair

Returns: A jSON string containing a status object.

7.4 View Job Pair Output

URL: services/jobs/pairs/{id}/stdout/{stageNumber}

URL Variables

{id} : Integer – The ID of the job pair to get the output of

{stageNumber} : Integer – Which stage number of the pair to get the results of. If the pair was not created with a solver pipeline, then this should be 1. If it was created with a solver pipeline, then it is the relevant stage number to retrieve.

Method: GET

Parameter Encoding: N/A

Description: Retrieves the job pair output of a single stage of the given job pair

Returns: A plaintext string containing the output of the given pair. Returns "not available" if the output could not be found or if you do not have permission to view the given pair.

7.5 View Job Pair Log

URL: services/jobs/pairs/{id}/log

URL Variables

{id} : Integer – The ID of the job pair to get the log of

Method: GET

Parameter Encoding: N/A

Description: Retrieves the job pair log of the single job pair given.

Returns: A plaintext string containing the job pair log of the given pair. Returns "not available" if the log could not be found or if you do not have permission to view the log of the given pair.

8 Websites

8.1 Add a New Website

URL: services/website/add/{type}/{id}

URL Variables

{type} : String – The type of primitive to add a website to. Can be "user", "space", or "solver"

{id} : Integer – The ID of the primitive to add a website too

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

name : String – The name of the website

url : String – The URL to the website

Description: Associates the given website with the specified primitive

Returns: A jSON string containing a status object

8.2 Delete a Website

URL: services/websites/delete/{websiteId}

URL Variables

{websiteId} : Integer – The ID of the website

Method: POST

Parameter Encoding: N/A

Description: Deletes the website with the given ID

Returns: A jSON string containing a status object

9 Users

9.1 Add Users to Space

URL: services/spaces/{spaceId}/add/user

URL Variables

{spaceId} : Integer – The ID of the destination space for all the users

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of user IDs to use

copyToSubspaces : Boolean – Whether to copy the users to every space in the hierarchy rooted at the given space.

Description: Associates all of the given users with the given space

Returns: A jSON string containing a status object.

9.2 Remove Users From Space

URL: services/remove/user/{spaceId}

URL Variables

{spaceId} : Integer – The ID of the space to remove users from

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

selectedIds[] : [integer] – The list of user IDs to use

hierarchy : Boolean – If true, removes the given users from the entire space hierarchy. Otherwise, just removes them from the given space.

Description: Removes all of the given users from the given space or space hierarchy, depending on the value of the hierarchy parameter.

Returns: A jSON string containing a status object.

9.3 Leave Space

URL: services/leave/space/{spaceId}

URL Variables

{spaceId} : Integer – The ID of the space to leave

Method: POST

Parameter Encoding: N/A

Description: Removes you from the given space and all of its subspaces.

Returns: A jSON string containing a status object.

9.4 Edit User Permissions in a Space

URL: services/space/{spaceId}/edit/perm/{userId}

URL Variables

{spaceId} : Integer – The ID of the relevant space

{userId} : Integer – The ID of the user to change permissions for

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

all permissions parameters (see the permissions section)

Description: Updates the permissions of the given user in a space.

Returns: A jSON string containing a status object.

9.5 Edit User Permissions in a Space Hierarchy

URL: services/space/{spaceId}/edit/perm/hier/{userId}

URL Variables

{spaceId} : Integer – The ID of the space that is at the root of the relevant hierarchy

{userId} : Integer – The ID of the user to change permissions for,

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

all permissions parameters (see the permissions section)

Description: Updates the permissions of the given user in a space hierarchy rooted at the given space.

Returns: A jSON string containing a status object.

10 Account

10.1 Logout

URL: services/session/logout

Method: POST

Parameter Encoding: N/A

Description: Logs the requesting user out of the system

Returns: A jSON string containing a status object

10.2 Request to Join Community

URL: add/to_community/request

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

cm : Integer – ID of the community (the space ID).

msg : String – Message explaining your motivation for joining the community.

Description: Sends a request to join a new community to the leaders of that community.

Returns: A redirect to a new webpage stating whether your message was sent.

10.3 Edit Account Data

URL: services/edit/user/{attr}/{userId}/{val}

URL Variables

{attr} : String – The type of attribute to update. Can be “firstname” “lastname” “institution”, “pagesize”

{userId} : Integer – Your user ID

{val} : <String or Integer, depending on attr> – The new value to set

Method: POST

Parameter Encoding: N/A

Description: Updates a field of profile information (like first name, last name, and so on). Note that the “pagesize” attribute refers to the default number of rows that will be displayed in the data tables displayed on Starexec.

Returns: A jSON string containing a status object.

10.4 Change Password

URL: services/edit/user/password/{userId}

URL Variables

{userId} : Integer – Your user ID

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

current : String – Your current password.

newPass : String – The new password you want.

Confirm : String – The new password you want, repeated for security

Description: Changes the password you use to log into Starexec

Returns: A jSON string containing a status object.

Return Cookie

JSESSIONID : String – The session ID for the log in. Note that you will need to include this unique key for all transmissions to the secure parts of Starexec.

10.5 Get User ID

URL: services/users/getid

Method: POST

Parameter Encoding: N/A

Description: Retrieves your user ID.

Returns: A jSON string containing your user ID.

10.6 Create or Modify Default Settings Profile

URL: add/profile

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

postp : Integer – The ID of the post processor to use in the settings. -1 to indicate none.

prep : Integer – The ID of the pre processor to use in the settings. -1 to indicate none.

benchp : Integer – The ID of the benchmark processor to use in the settings. -1 to indicate none.

solver : Integer – The ID of the solver to use in the settings. -1 to indicate none.

bench : Integer – The ID of the benchmark to use in the settings. -1 to indicate none.

cpu : Integer – The cpu timeout for the profile.

wall : Integer – the wallclock timeout for the profile.

mem : Float – The maximum memory for the profile, in gigabytes.

dep : Boolean – Whether to enable dependencies by default in the profile

+ exactly 1 of the following two parameters

name : String – The name of the new setting profile. Use to create a new profile.

settingId : Integer – The ID of an existing settings profile that you own. Use to modify the profile with new values.

Description: Given a set of settings for a profile, either creates a new profile with those settings (if name is given) or modifies an existing profile with new values (if settingId is given).

Returns: An HTTP message with 200 status on success, and an HTTP message with an error status on failure.

10.7 Set Default Settings Profile

URL: services/set/defaultSettings/{id}

URL Variables

{id} : Integer – The ID of the settings profile to use

Method: POST

Parameter Encoding: N/A

Description: Sets the given settings profile as your default profile, meaning the one that is loaded automatically upon visiting pages like the job creation page.

Returns: A jSON string containing a status object.

10.8 Edit Default Settings Profile

URL: services/edit/defaultSettings/{attr}/{id}

URL Variables

{id} : Integer – The ID of the settings profile to edit

{attr} : String – The attribute to edit. Can be “PostProcess”, “BenchProcess”, “CpuTimeout”, “ClockTimeout”, “DependenciesEnabled”, “defaultbenchmark”, “defaultsolver”, “MaxMem”, “PreProcess”

Method: POST

Parameter Encoding: application/x-www-form-urlencoded

Parameters

val : <String or Int, depending on choice of {attr}> -- The new value to use for the given attribute

Description: Edits an existing settings profile by using the given new value for the given attribute

Returns: A jSON string containing a status object.

10.9 Delete Default Settings Profile

URL: services/delete/defaultSettings/{id}

URL Variables

{id} : Integer – The ID of the settings profile to delete

Method: POST

Parameter Encoding: N/A

Description: Deletes the settings profile with the given ID

Returns: A jSON string containing a status object.