SSR slicer modify

FR-N-S087087:slicer\_cli\_web\_SSR miaot2$ git diff --color | cat

**diff --git a/Dockerfile b/Dockerfile**

**index b8230bb..f17de70 100644**

**--- a/Dockerfile**

**+++ b/Dockerfile**

@@ -81,8 +81,8 @@ RUN cd $BUILD\_PATH && \

# Install ctk-cli

RUN conda install --yes -c cdeepakroy ctk-cli=1.4.1

-# copy slicer\_cli\_web files

-ENV slicer\_cli\_web\_path=$BUILD\_PATH/slicer\_cli\_web

-RUN mkdir -p $slicer\_cli\_web\_path

-COPY . $slicer\_cli\_web\_path/

-WORKDIR $slicer\_cli\_web\_path/server

\ No newline at end of file

+# copy slicer\_cli\_web\_SSR files

+ENV slicer\_cli\_web\_SSR\_path=$BUILD\_PATH/slicer\_cli\_web\_SSR

+RUN mkdir -p $slicer\_cli\_web\_SSR\_path

+COPY . $slicer\_cli\_web\_SSR\_path/

+WORKDIR $slicer\_cli\_web\_SSR\_path/server

\ No newline at end of file

**diff --git a/plugin.json b/plugin.json**

**index 67a0c71..9256f91 100644**

**--- a/plugin.json**

**+++ b/plugin.json**

@@ -1,5 +1,5 @@

{

- "name": "slicer\_cli\_web",

+ "name": "slicer\_cli\_web\_SSR",

"description": "A girder plugin for exposing slicer CLIs over the web",

"version": "0.2.0",

"dependencies": ["worker", "jobs"],

**diff --git a/plugin\_tests/docker\_test.py b/plugin\_tests/docker\_test.py**

**index bec2cfc..bc26ee8 100644**

**--- a/plugin\_tests/docker\_test.py**

**+++ b/plugin\_tests/docker\_test.py**

@@ -31,7 +31,7 @@ TIMEOUT = 180

def setUpModule():

- base.enabledPlugins.append('slicer\_cli\_web')

+ base.enabledPlugins.append('slicer\_cli\_web\_SSR')

base.startServer()

global JobStatus

from girder.plugins.jobs.constants import JobStatus

**diff --git a/plugin\_tests/example\_test.py b/plugin\_tests/example\_test.py**

**index 8a8c2d4..70c9edf 100644**

**--- a/plugin\_tests/example\_test.py**

**+++ b/plugin\_tests/example\_test.py**

@@ -25,7 +25,7 @@ from tests import base

# boiler plate to start and stop the server

def setUpModule():

- base.enabledPlugins.append('slicer\_cli\_web')

+ base.enabledPlugins.append('slicer\_cli\_web\_SSR')

base.startServer()

**diff --git a/plugin\_tests/import\_test.py b/plugin\_tests/import\_test.py**

**index bd5be46..675b95b 100644**

**--- a/plugin\_tests/import\_test.py**

**+++ b/plugin\_tests/import\_test.py**

@@ -22,7 +22,7 @@ from tests import base

# boiler plate to start and stop the server if needed

def setUpModule():

- base.enabledPlugins.append('slicer\_cli\_web')

+ base.enabledPlugins.append('slicer\_cli\_web\_SSR')

base.startServer()

@@ -34,17 +34,17 @@ def tearDownModule():

class ImportPackageTest(base.TestCase):

def test\_slicer\_cli\_web(self):

- from girder.plugins import slicer\_cli\_web # noqa

+ from girder.plugins import slicer\_cli\_web\_SSR # noqa

def test\_rest\_slicer\_cli(self):

- from girder.plugins.slicer\_cli\_web import rest\_slicer\_cli # noqa

+ from girder.plugins.slicer\_cli\_web\_SSR import rest\_slicer\_cli # noqa

def test\_docker\_image\_model(self):

from girder.models.model\_base import ModelImporter

raised = False

try:

- ModelImporter.model('docker\_image\_model', 'slicer\_cli\_web') # noqa

+ ModelImporter.model('docker\_image\_model', 'slicer\_cli\_web\_SSR') # noqa

except Exception as e:

raised = True

print str(e)

@@ -52,7 +52,7 @@ class ImportPackageTest(base.TestCase):

self.assertFalse(raised, 'docker\_image\_model import failed')

def test\_docker\_resource(self):

- from girder.plugins.slicer\_cli\_web import docker\_resource # noqa

+ from girder.plugins.slicer\_cli\_web\_SSR import docker\_resource # noqa

def test\_cli\_list\_entrypoint(self):

- from girder.plugins.slicer\_cli\_web import cli\_list\_entrypoint # noqa

+ from girder.plugins.slicer\_cli\_web\_SSR import cli\_list\_entrypoint # noqa

**diff --git a/plugin\_tests/parseSpec.js b/plugin\_tests/parseSpec.js**

**index afdb888..bce7960 100644**

**--- a/plugin\_tests/parseSpec.js**

**+++ b/plugin\_tests/parseSpec.js**

@@ -1,8 +1,8 @@

-girderTest.importPlugin('jobs', 'worker', 'slicer\_cli\_web');

+girderTest.importPlugin('jobs', 'worker', 'slicer\_cli\_web\_SSR');

var parser;

girderTest.promise.done(function () {

- parser = girder.plugins.slicer\_cli\_web.parser;

+ parser = girder.plugins.slicer\_cli\_web\_SSR.parser;

});

describe('XML Schema parser', function () {

**diff --git a/plugin\_tests/widgetSpec.js b/plugin\_tests/widgetSpec.js**

**index dd763bb..c734c70 100644**

**--- a/plugin\_tests/widgetSpec.js**

**+++ b/plugin\_tests/widgetSpec.js**

@@ -1,8 +1,8 @@

-girderTest.importPlugin('jobs', 'worker', 'slicer\_cli\_web');

+girderTest.importPlugin('jobs', 'worker', 'slicer\_cli\_web\_SSR');

var slicer;

girderTest.promise.done(function () {

- slicer = girder.plugins.slicer\_cli\_web;

+ slicer = girder.plugins.slicer\_cli\_web\_SSR;

});

describe('widget model', function () {

**diff --git a/server/\_\_init\_\_.py b/server/\_\_init\_\_.py**

**index d9ec8e8..1f9dec1 100644**

**--- a/server/\_\_init\_\_.py**

**+++ b/server/\_\_init\_\_.py**

@@ -31,7 +31,7 @@ def load(info):

info['apiRoot'].slicer\_cli\_web = resource

dockerImageModel = ModelImporter.model('docker\_image\_model',

- 'slicer\_cli\_web')

+ 'slicer\_cli\_web\_SSR')

dockerCache = dockerImageModel.loadAllImages()

genRESTEndPointsForSlicerCLIsInDockerCache(resource, dockerCache)

**diff --git a/server/docker\_resource.py b/server/docker\_resource.py**

**index 3307fa6..cd66f15 100644**

**--- a/server/docker\_resource.py**

**+++ b/server/docker\_resource.py**

@@ -58,7 +58,7 @@ class DockerResource(Resource):

def getDockerImages(self, params):

dockermodel = ModelImporter.model('docker\_image\_model',

- 'slicer\_cli\_web')

+ 'slicer\_cli\_web\_SSR')

dockerCache = dockermodel.loadAllImages()

cache = dockerCache.getImages()

@@ -147,7 +147,7 @@ class DockerResource(Resource):

"""

dockermodel = ModelImporter.model('docker\_image\_model',

- 'slicer\_cli\_web')

+ 'slicer\_cli\_web\_SSR')

try:

dockermodel.removeImages(names)

@@ -202,7 +202,7 @@ class DockerResource(Resource):

self.requireParams(('name',), params)

nameList = self.parseImageNameList(params['name'])

docker\_image\_model = ModelImporter.model('docker\_image\_model',

- 'slicer\_cli\_web')

+ 'slicer\_cli\_web\_SSR')

return docker\_image\_model.putDockerImage(nameList, self.jobType, True)

def storeEndpoints(self, imgName, cli, operation, argList):

@@ -272,7 +272,7 @@ class DockerResource(Resource):

if job['type'] == self.jobType and job['status'] == JobStatus.SUCCESS:

# remove all previous endpoints

dockermodel = ModelImporter.model('docker\_image\_model',

- 'slicer\_cli\_web')

+ 'slicer\_cli\_web\_SSR')

cache = dockermodel.loadAllImages()

self.deleteImageEndpoints()

**diff --git a/server/image\_job.py b/server/image\_job.py**

**index 9789410..00f8504 100644**

**--- a/server/image\_job.py**

**+++ b/server/image\_job.py**

@@ -148,7 +148,7 @@ def jobPullAndLoad(job):

cache, loadingError = LoadMetadata(jobModel, job, docker\_client,

pullList, loadList, notExistSet)

imageModel = ModelImporter.model('docker\_image\_model',

- 'slicer\_cli\_web')

+ 'slicer\_cli\_web\_SSR')

imageModel.saveAllImgs(cache)

if errorState is False and loadingError is False:

**diff --git a/server/models/docker\_image\_model.py b/server/models/docker\_image\_model.py**

**index 1b98417..bd6f7c7 100644**

**--- a/server/models/docker\_image\_model.py**

**+++ b/server/models/docker\_image\_model.py**

@@ -88,7 +88,7 @@ class DockerImageModel(AccessControlledModel):

pullList.append(name)

job = jobModel.createLocalJob(

- module='girder.plugins.slicer\_cli\_web.image\_job',

+ module='girder.plugins.slicer\_cli\_web\_SSR.image\_job',

function='jobPullAndLoad',

kwargs={

'pullList': pullList,

@@ -213,7 +213,7 @@ class DockerImageModel(AccessControlledModel):

job = jobModel.createLocalJob(

- module='girder.plugins.slicer\_cli\_web.image\_job',

+ module='girder.plugins.slicer\_cli\_web\_SSR.image\_job',

function='deleteImage',

kwargs={

**diff --git a/server/rest\_slicer\_cli.py b/server/rest\_slicer\_cli.py**

**index a7fbdce..dff21da 100644**

**--- a/server/rest\_slicer\_cli.py**

**+++ b/server/rest\_slicer\_cli.py**

@@ -34,19 +34,22 @@ \_SLICER\_TO\_GIRDER\_WORKER\_TYPE\_MAP = {

'file': 'string',

'directory': 'string',

'image': 'string'

+

}

\_SLICER\_TYPE\_TO\_GIRDER\_MODEL\_MAP = {

'image': 'file',

'file': 'file',

'item': 'item',

- 'directory': 'folder'

+# 'directory': 'folder'

+ 'directory': 'item'

}

\_SLICER\_TYPE\_TO\_GIRDER\_INPUT\_SUFFIX\_MAP = {

'image': '\_girderFileId',

'file': '\_girderFileId',

'item': '\_girderItemId',

- 'directory': '\_girderFolderId',

+# 'directory': '\_girderFolderId'

+ 'directory': '\_girderItemId'

}

\_worker\_docker\_data\_dir = constants.DOCKER\_DATA\_VOLUME

**diff --git a/web\_client/collections/WidgetCollection.js b/web\_client/collections/WidgetCollection.js**

**index 40dcbd5..63b039e 100644**

**--- a/web\_client/collections/WidgetCollection.js**

**+++ b/web\_client/collections/WidgetCollection.js**

@@ -21,6 +21,9 @@ var WidgetCollection = Backbone.Collection.extend({

case 'item':

params[m.id + '\_girderItemId'] = m.value().id;

break;

+ case 'directory':

+ params[m.id + '\_girderItemId'] = m.value().id;

+ break;

case 'new-file':

params[m.id + '\_girderFolderId'] = m.value().get('folderId');

params[m.id + '\_name'] = m.value().get('name');

**diff --git a/web\_client/main.js b/web\_client/main.js**

**index bb31603..cdf52cf 100644**

**--- a/web\_client/main.js**

**+++ b/web\_client/main.js**

@@ -1,4 +1,4 @@

import { registerPluginNamespace } from 'girder/pluginUtils';

-import \* as slicer from 'girder\_plugins/slicer\_cli\_web';

-registerPluginNamespace('slicer\_cli\_web', slicer);

+import \* as slicer from 'girder\_plugins/slicer\_cli\_web\_SSR';

+registerPluginNamespace('slicer\_cli\_web\_SSR', slicer);

**diff --git a/web\_client/templates/itemSelectorWidget.pug b/web\_client/templates/itemSelectorWidget.pug**

**index 51034e8..a8ab06a 100644**

**--- a/web\_client/templates/itemSelectorWidget.pug**

**+++ b/web\_client/templates/itemSelectorWidget.pug**

@@ -18,9 +18,9 @@

label.control-label(for='s-new-file-name') Item name

input#s-new-file-name.form-control(type='text', value=value, placeholder='Enter an item name...')

- if type === 'directory' || type === 'new-file'

- .modal-footer

- a.btn.btn-small.btn-default(data-dismiss="modal") Cancel

- button.s-select-button.btn.btn-small.btn-primary(type="submit")

- i.icon-edit

- | Save

+// if type === 'directory' || type === 'new-file'

+// .modal-footer

+// a.btn.btn-small.btn-default(data-dismiss="modal") Cancel

+// button.s-select-button.btn.btn-small.btn-primary(type="submit")

+// i.icon-edit

+// | Save

**diff --git a/web\_client/views/ItemSelectorWidget.js b/web\_client/views/ItemSelectorWidget.js**

**index 228bbaa..4dc4a50 100644**

**--- a/web\_client/views/ItemSelectorWidget.js**

**+++ b/web\_client/views/ItemSelectorWidget.js**

@@ -61,10 +61,12 @@ var ItemSelectorWidget = View.extend({

switch (this.model.get('type')) {

case 'item':

+

this.model.set({

path: this.\_path(),

value: item

});

+

this.trigger('g:saved');

this.$el.modal('hide');

break;

@@ -109,6 +111,16 @@ var ItemSelectorWidget = View.extend({

}, this)).fetch();

this.$el.modal('hide');

break;

+

+ case 'directory':

+ this.model.set({

+ path: this.\_path(),

+ value: item

+ });

+

+ this.trigger('g:saved');

+ this.$el.modal('hide');

+ break;

}

},