PyDynamicReporting cheat sheet

Version: 0.4.0 (stable)

Launch Dynamic Reporting Service

Launch and stop a local instance of the Ansys Dynamic Reporting Service on a new database:

```
import ansys.dynamicreporting.core as adr
db_dir = r'C:\tmp\my_local_db_directory'
ansys_ins = r'C:\Program Files\Ansys Inc\v241'
adr_service = adr.Service(
  ansys_installation=ansys_ins, db_directory=db_dir)
session_guid = adr_service.start(create_db=True)
# Stop the service
adr_service.stop()
```

Connect to an already running Ansys Dynamic Reporting service on port 8000:

```
import ansys.dynamicreporting.core as adr
ansys_ins = r'C:\Program Files\Ansys Inc\v241'
adr_service = adr.Service(
  ansys_installation=ansys_ins)
adr_service.connect(url='http://localhost:8000')
```

Create items

Create items of different types in the database:

```
# Create text item
my_text = adr_service.create_item(obj_name='Text')
my_text.item_text = "<h1>Simple Title</h1>Abc..."
# Create table item
import numpy as np
my_table = adr_service.create_item(obj_name='Table')
my_table.table_dict["rowlbls"] = ["Row 1", "Row 2"]
my_table.item_table = np.array([
  ["1", "2", "3", "4", "5"],
  ["1", "4", "9", "16", "25"]], dtype="|S20")
# Create image item
img = adr_service.create_item(obj_name='Image')
img.item_image = r'C:\tmp\test_image.png'
# Create 3D item
scene = adr_service.create_item(obj_name='3D Scene')
scene.item_scene = r'C:\tmp\test_scene.avz'
# Create a tree item via a dictionary
leaves = []
for i in range(5):
  leaves.append({"key": "leaves", "name": f"Leaf {i}",
    "value": i})
children = []
children.append({"key": "child", "name": "Boolean
    example", "value": True})
children.append({
  "key": "child_parent",
  "name": "A child parent",
  "value": "Parents can have values",
  "children": leaves,
  "state": "collapsed"})
tree = []
tree.append(
  {"key": "root", "name": "Top Level", "value": None,
  "children": children, "state": "expanded"})
my_tree = adr_service.create_item(obj_name="Tree")
my_tree.item_tree = tree
```

Set plot properties

Display a table item as a plot:

```
# Set visualization to be a plot instead of a table
my_table.plot = 'line'
# Set X axis and axis formatting
my_table.xaxis = 'Row 1'
my_table.format = 'floatdot1'
```

Tag items

Set tags on an item:

```
my_text.set_tags("tag1=one tag2=two tag3=three")
```

Add or remove tags on an item:

```
my_text.add_tag(tag='tag4', value='four')
my_text.rem_tag("tag1")
```

Create report templates

You use the low-level API for Ansys Dynamic Reporting to create report templates. First define the server object:

```
server = adr_service.serverobj
```

Then use the low-level API to create a report template:

```
template_0=server.create_template(name="My Report",
  parent=None, report_type="Layout:basic")
server.put_objects(template_0)

template_1=server.create_template(name="Intro",
  parent=template_0, report_type="Layout:panel")
template_1.set_filter("A|i_type|cont|html,string;")
server.put_objects(template_0)
server.put_objects(template_1)

template_2=server.create_template(name="Plot",
  parent=template_0, report_type="Layout:panel")
template_2.set_filter("A|i_type|cont|table;")
server.put_objects(template_2)
server.put_objects(template_0)
```

Visualize items and reports

Search for text items:

```
all_items = adr_service.query()
only_text_items = adr_service.query(filter=
  "A|i_type|cont|html,string")
```

Visualize the first item returned:

```
only_text_items[0].visualize()
```

Visualize all items in the report, listed one after the other:

```
adr_service.visualize_report()
```

List all reports:

```
all_reports = adr_service.get_list_reports()
```

View a report with a specific name:

```
report_by_name = adr_service.get_report(
  report_name='My Report')
report_by_name.visualize()
```

Get URLs for items and reports

Get the URL from the first item returned:

```
only_text_items[0].url
```

Get the URL of a report:

```
report_by_name.get_url()
```

References from PyAnsys documentation

- Getting started
- API reference
- Examples