

Documentation for Snap-in Assignment

This documentation covers the three Python files, designed to interact with the DevRev API for work item management.

1) work_item.py

Purpose: Creates a new work item in DevRev.

Key Features:

- Uses the DevRev API token for authentication.
- Accepts an artifact data dictionary with details of the work item.
- Sends a POST request to the DevRev API endpoint for work item creation.
- Prints the artifact ID if creation is successful.
- Logs error messages if the creation fails.

Dependencies:

- requests library

2) hello_world.py

Purpose: Creates a simple "Hello World" snap-in in DevRev.

Key Features:

Defines a function `create_snap_in` that sends a POST request to the DevRev API to create a snap-in.

The `snap_in_data` dictionary contains the configuration for the "Hello World" snap-in:

Name: "HelloWorld"

Description: "Simple 'Hello World' snap-in"

Trigger Events: "workItem_created"

Actions: Sends a message "Hello, World!" upon trigger

Uses the DevRev API token for authentication.

Logs success/error messages.

Dependencies:

requests library

json library

3. clone_work_item.py

Purpose: Creates a snap-in that clones a work item in DevRev.

Key Features:

Defines a function create_snap_in that sends a POST request to the DevRev API to create a snap-in.

The snap_in_data dictionary contains the configuration for the "CloneWorkItem" snap-in:

Name: "CloneWorkItem"

Description: "Clones a work item"

TriggerEvents: "workItem_created"

Actions:

Creates a new work item with copied data (summary, description, status, type, project).

Sends a message confirming the clone and providing the URL of the copied work item.

Uses DevRev API token for authentication.

Logs success/error messages.

Dependencies:

requests library

json library