Quera

Module to host QueraBackend class that hosts different functionalities representing the backend of the QuEra system(s).

QuEraBackend

Bases: `SubmissionBackend`

Class to represent QuEra backend.

Attributes:

Name	Type	Description
`api_hostname`	`str`	API of the host.
`qpu_id`	`str`	QPU id.
`api_stage`	`str`	API version. Defaults to "v0"
`proxy`	`Optional[str]`	proxy for Quera backend. Defaults to `None`.
`region`	`Optional[str]`	Sigv4Request argument for region of QuEra backend. Defaults to `None`.
`access_key`	`Optional[str]`	Sigv4Request argument representing the access key required to access QuEra backend. Defaults to `None`.
`secret_key`	`Optional[str]`	Sigv4Request argument representing the secret key required to access QuEra backend. Defaults to `None`.
`session_token`	`Optional[str]`	Sigv4Request argument representing the session token of the QuEra backend. Defaults to `None`.

queue_api `property`

```
queue_api
```

TODO: Document

cancel_task

```
cancel_task(task_id)
```

Cancels the task submitted to the QuEra backend.

Parameters:

Name	Type	Description	Default
`task_id`	`str`	task id after executing program on the QuEra backend.	*required*

Source code in `src\bloqade\submission\quera.py`

```
120  def cancel_task(self, task_id: str):
121      """Cancels the task submitted to the QuEra backend.
122
123      Args:
124          task_id (str): task id after executing program on the QuEra backend.
125      """
126      self.queue_api.cancel_task_in_queue(task_id)
```

get_capabilities

```
get_capabilities(use_experimental=False)
```

Get the capabilities of the QuEra backend.

Parameters:

Name	Type	Description	Default
`use_experimental`	`bool`	Whether to use experimental capabilities of the QuEra system. Defaults to `False`.	`False`

Returns:

Name	Type	Description
`capabilities`	`QuEraCapabilities`	capabilities of the selected QuEra backend.

Source code in `src\bloqade\submission\quera.py`

```
76  def get_capabilities(self, use_experimental: bool = False) ->
77  QuEraCapabilities:
78      """Get the capabilities of the QuEra backend.
79
80      Args:
81          use_experimental (bool): Whether to use experimental capabilities of
82              the QuEra system. Defaults to `False`.
83
84      Returns:
85          capabilities (QuEraCapabilities): capabilities of the selected
86              QuEra backend.
87      """
88      try:
89          return QuEraCapabilities(**self.queue_api.get_capabilities())
90      except BaseException:
            return super().get_capabilities(use_experimental)
```

submit_task

```
submit_task(task_ir)
```

Submit the task to the QuEra backend.

Parameters:

Name	Type	Description	Default
`task_ir`	`QuEraTaskSpecification`	task IR to be executed on the QuEra backend.	*required*

Returns:

Name	Type	Description
`task_id`	`str`	Task id as a result of executing IR on the QuEra backend.

Source code in `src\bloqade\submission\quera.py`

```
92  def submit_task(self, task_ir: QuEraTaskSpecification) -> str:
93      """Submit the task to the QuEra backend.
94
95      Args:
96          task_ir (QuEraTaskSpecification): task IR to be
97              executed on the QuEra backend.
98
99      Returns:
100         task_id (str): Task id as a result of executing
101             IR on the QuEra backend.
102     """
103     return self.queue_api.submit_task(
104         task_ir.json(by_alias=True, exclude_none=True, exclude_unset=True)
105     )
```

task_results

```
task_results(task_id)
```

Get the status of the task submitted to the QuEra backend by using the task id.

Parameters:

Name	Type	Description	Default
`task_id`	`str`	task id after executing program on the QuEra	*required*

Name	Type	Description	Default
		backend.	

Returns:

Name	Type	Description
`task_result`	`QuEraTaskResults`	Final result of the task by using the task id.

Source code in `src\bloqade\submission\quera.py`

```
107  def task_results(self, task_id: str) -> QuEraTaskResults:
108      """Get the status of the task submitted to the QuEra backend
109      by using the task id.
110
111      Args:
112          task_id (str): task id after executing program on the QuEra backend.
113
114      Returns:
115          task_result (QuEraTaskResults):
116              Final result of the task by using the task id.
117      """
118      return QuEraTaskResults(**self.queue_api.poll_task_results(task_id))
```

task_status

```
task_status(task_id)
```

Get the status of the task submitted to the QuEra backend by using the task id.

Parameters:

Name	Type	Description	Default
`task_id`	`str`	task id after executing program on the QuEra backend.	*required*

Returns:

Name	Type	Description
`task_status`	`QuEraTaskStatusCode`	status of the task by using the task id.

Source code in `src\bloqade\submission\quera.py`

```
128  def task_status(self, task_id: str) -> QuEraTaskStatusCode:
129      """Get the status of the task submitted to the QuEra backend
130      by using the task id.
131
132      Args:
133          task_id (str): task id after executing program on the QuEra backend.
134
135      Returns:
136          task_status (QuEraTaskStatusCode): status of the task by using the
137  task id.
138      """
139      return_body = self.queue_api.get_task_status_in_queue(task_id)
         return QuEraTaskStatusCode(return_body)
```

update_credential

```
update_credential(
    access_key=None, secret_key=None, session_token=None
)
```

Update the credentials

Parameters:

Name	Type	Description	Default
`access_key`	`Optional[str]`	Sigv4Request argument representing the access key required to access QuEra backend. Defaults to `None`	`None`
`secret_key`	`Optional[str]`	Sigv4Request argument representing the secret key required to access QuEra backend. Defaults to `None`	`None`

Name	Type	Description	Default
`session_token`	`Optional[str]`	Sigv4Request argument representing the session token of the QuEra backend. Defaults to `None`	`None`

Source code in `src\bloqade\submission\quera.py`

```
158  def update_credential(
159      self, access_key: str = None, secret_key: str = None, session_token: str
160  = None
161  ):
162      """Update the credentials
163
164      Args:
165          access_key (Optional[str]): Sigv4Request argument representing the
166  access
167              key required to access QuEra backend. Defaults to `None`
168          secret_key (Optional[str]): Sigv4Request argument representing the
169  secret
170              key required to access QuEra backend. Defaults to `None`
171          session_token (Optional[str]): Sigv4Request argument representing the
172              session token of the QuEra backend. Defaults to `None`
173      """
174      if secret_key is not None:
175          self.secret_key = secret_key
176      if access_key is not None:
          self.access_key = access_key
      if session_token is not None:
          self.session_token = session_token
```

validate_task

```
validate_task(task_ir)
```

Validates the task submitted to the QuEra backend.

Parameters:

Name	Type	Description	Default
`task_ir`	`QuEraTaskSpecification`	task IR to be executed on the	*required*

Name	Type	Description	Default
		QuEra backend.	

Raises:

Type	Description
`ValidationError`	For tasks that fail validation.

Source code in `src\bloqade\submission\quera.py`

```
141     def validate_task(self, task_ir: QuEraTaskSpecification):
142         """Validates the task submitted to the QuEra backend.
143 
144         Args:
145             task_ir (QuEraTaskSpecification): task IR to be
146                 executed on the QuEra backend.
147 
148         Raises:
149             ValidationError: For tasks that fail validation.
150         """
151         try:
152             self.queue_api.validate_task(
153                 task_ir.json(by_alias=True, exclude_none=True,
154 exclude_unset=True)
155             )
156         except self.queue_api.ValidationError as e:
            raise ValidationError(str(e))
```