

1 Schnittstellen zwischen Frontend und Backend

Übersicht

Schnittstelle	Richtung	Format	Typ
Upload Encoding	FE → BE	JSON	POST
Start Benchmark	FE → BE	JSON	POST
Ergebnisabruf	BE → FE	JSON	GET
Reference Data	BE → FE	JSON	GET
Status Benchmark	BE → FE	JSON	GET

- **Encoding Upload API**

FE → BE

CRUD: /api/encoding

Überträgt hochgeladene oder im Editor erstellte Quantencodierungen im JSON-Format.

Beispiel:

```
[
  {"gate": "RY", "wires": [0], "params": ["theta_1"]},
  {"gate": "CNOT", "wires": [0, 1]},
  {"gate": "RZ", "wires": [1], "params": ["x_1"]},
  {"gate": "RZ", "wires": [1], "params": [0.33]}
]
```

Response:

```
{"id": 123}
```

- **Run Benchmark Request API**

FE → BE

CRUD: /api/run

Startet die Evaluation für ein Encoding in Kombination mit einem gewählten Datensatz und Ansatz.

Response:

```
{"id": 123}
```

- **Benchmark Result API**

BE → FE

CRUD: /api/result

Gibt Ergebnisdaten wie Accuracy, Loss und Circuit Depth im JSON-Format zurück. Diese Daten werden im Dashboard visualisiert.

Response:

```
{
  "results":
  [
    {
      "encoding_id": 2,
      "ansatz_id": 1,
      "data_id": 1,
      "loss": 123,
      "accuracy": 123
    },
    ...
  ],
  "encodings":
  {
    2: {"depth": 5, "name": "Basis encoding", "description": "Lorem ipsum" },
    ...
  },
  "ansaetze":
  {
    2: {"depth": 5, "name": "Basis encoding", "description": "Lorem ipsum" },
    ...
  }
}
```

- **ReferenceData API**

BE → FE

CRUD: /api/dataset

Liefert Informationen über verfügbare Datasets, Ansätze und Baseline-Encodings zur Anzeige im Frontend.

Beispiel:

```
{
  "encodings":
  {
    2: {"depth": 5, "name": "Basis encoding", "description": "Lorem ipsum", "circuit": {...} },
    ...
  },
  "ansaetze":
  {
    2: {"depth": 5, "name": "Random Layer", "description": "Lorem ipsum", "circuit": {...} },
    ...
  },
  "data":
  {
    2: {"depth": 5, "name": "Digits", "description": "Lorem ipsum"},
    ...
  }
}
```