Versionierung

verschiedene API-Versionen

Api Versionierung

Die API-Versionierung ermöglicht es, das Verhalten zwischen verschiedenen Clients zu ändern. Das REST-Framework bietet eine Reihe verschiedener Versionierungsschemata.

Die Versionierung wird durch die eingehende Client-Anfrage bestimmt und kann entweder auf der Anfrage-URL oder auf den Anfrage-Headern basieren.

Versionierung nutzen

Um die Versionierung zu nutzen, kann in den Methoden der Serializer bzw. Views die aktuell genutzte Version ausgelesen werden.

Die Versionierung findet sich als Attribut des Request-Objekts:

self.request.version

Konfiguration

In den Settings des Restframeworks kann die Default-Versionierungsart angegeben werden. Diese gilt für das gesamte Projekt.

```
REST_FRAMEWORK = {
    'DEFAULT_VERSIONING_CLASS': 'rest_framework.versioning.NamespaceVersioning'
}
```

Der Defaultwert ist None, und damit nicht gegeben.

URL Pfad Versionierung

Bei der URL-Pfad Versionierung wird die Version in der URL abgebildet. Fehlt die Version, tritt ein Fehler auf.

api/v1/events

in den URL-Patterns muss dafür Sorge tragen werden, dass diese Version übernommen und erkannt wird.

```
urlpattern = [
    re_path(r"^api/(?P<version>(v1|v2))/events/$, ..., ),
]
```

Der reguläre Ausdruck findet den String v1 oder v2 und speichert ihn in der Capturing Group version. Diese Version steht dann später im Code zur Verfügung.

Name Space Versionierung

Die Namespace-Versionierung ist sehr ähnlich zur URL-Pfad Versionierung. Hier ist der Namespace der include-Funktion ausschlaggebend. Für den User besteht zwischen Namespace-Versionierung und URL-Pfad Versionierung kein Unterschied.

```
urlpatterns = [
    re_path(r'^api/v1/events/', include('events.urls', namespace='v1')),
    re_path(r'^api/v2/events/', include('events.urls', namespace='v2'))
]
```

Accept Header Versionierung

Der Client muss bei dieser Art von Versionierung im Accept-Header die Version angeben, wenn er einen Request durchführt. Accept-Header Versionierung gilt als best practice, ist aber auch am aufwändigsten für den Client.

GET /events/ HTTP/1.1

Host: example.com

Accept: application/json; version=1.0