# REST (Representational State Transfer)

Before you familiarize yourself with the framework you have selected, we ask you to carry out the following work orders.

**1. Translate the following vocabulary into German. Add any words of your choice that might be useful, too.**

| | | | |
|---|---|---|---|
| development | Entwicklung | loop | Schleife |
| string | Zeichenkette | value | Wert |
| character | Zeichen | dictionary | Wörterbuch?! |
| application | Anwendung | library | Bibliothek |
| list | Liste | conclusion | Schlussfolgerung |
| to remove sth | etwas entfernen | inheritance | Vererbung |
| to append sth | etwas hinzufügen | interface | Schnittstelle |
| (to) reverse | umkehren, rückgängig machen | exception | Ausnahme |
| authentication | Authentifizierung | application programming interface | Programmierschnittstelle |
| to retrieve sth | etw. Wiederherstellen | relationship | Beziehung |
| conditional | bedingt | endpoint | Endpunkt |
| to execute sth | etwas ausführen | to host | Hosten, Gastgeber sein |
| constraint | Beschränkung | to expose | zeigen, darlegen |
| | | | |
| | | | |

**2. Read the following English text. Translate all unclear terms and answer the questions below.**

https://t1p.de/jjh7w

**Questions on the text**

1. What is an API?
2. What are the two most important features of an API?
3. What advantage does using an API have when it comes to using a program on different platforms like Windows and Android?
4. What is a REST API?
5. Name the main characteristics of a REST API.

**General questions**

1. What advantages does the use of a framework in software development have?
2. What does "stateless" mean in the context of an API?

1. Application Programming Interface: interface to an application is the code that another application can interact with.
2. Interfaces and Endpoints
3. its less redundant, you can use the same code, without having to run it on your own pc
4. A REST API is just an API that follows specific conventions and has specific characteristics
5. Uniform interface, Client-server, Stateless, Cacheable, Layered system

1. reuseability of the code, controlling the access of information given to the client
2. The server (API) doesn't store anything about previous client requests. Each client request is treated as a brand new client.

3. **Have a look at the HTTP request methods below and make sure that you know all of them.**

**Important HTTP Request methods**

| | HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource. Although they can also be nouns, these request methods are sometimes referred to as *HTTP verbs*. Each of them implements a different semantic, but some common features are shared by a group of them: e.g. a request method can be safe, idempotent, or cacheable. |
|---|---|
| GET | The GET method requests a representation of the specified resource. Requests using GET should only retrieve data. |
| POST | The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server. |
| PUT | The PUT method replaces all current representations of the target resource with the request payload. |
| DELETE | The DELETE method deletes the specified resource. |

If you're interested in further information, click here: RFC 9110: HTTP Semantics (rfc-editor.org)