# Resource-oriented Client Architecture (ROCA)

In this lesson, we'll learn all about ROCA.

WE'LL COVER THE FOLLOWING



Modularization and integration in the frontend have an impact on the architecture and the technologies in the frontend. SPAs (single page apps) are not well suited for integration in the frontend. Therefore, the question arises as to **which frontend architecture is better suited for integration**.

> **ROCA** (Resource-oriented Client Architecture) is an approach for implementing web applications. It focuses on established technologies such as HTML and leads to an architecture that comprises many benefits for frontend modularization and integration.

## ROCA principles #

ROCA has a few principles. Let's discuss each.

ROCA has a few principles. Let's discuss each.

## Adherence to REST principles

The **server adheres to the REST principles**:

1. All resources have an unambiguous URL.
2. Links to web pages can be sent by e-mail and then accessed from any browser if the necessary authorizations are given.
3. HTTP methods are used correctly. For example, GETs do not change data.
4. The server is stateless.

## Accessible information

**Resources identified by URLs can have other representations** besides HTML, such as JSON or XML. This means that the data is not only available to people, but also to applications.

## Logic on server

**All logic is on the server**. Accordingly, JavaScript on the client-side only serves to optimize the user interface. Not just browsers but also other clients should be able to access the system.

Logic on the server is desirable for security reasons because the client could be manipulated. Changing the logic is also easier because it is implemented in one location making it unnecessary to update a large number of clients.

## Authentication in HTTP request

The **authentication information is included in the HTTP request**. HTTP basic, digest, client certificates, or cookies can be used for this purpose. In this way, authentication and authorization can take place solely on the basis of information contained in the HTTP request. Therefore, no server-side session is required for the authentication information.

## Cookies

Cookies may be used only for:

1. Authentication
2. Tracking
3. As protection against cross-site request forgery

Therefore, they may not contain business information.

## No server-side sessions

**There must not be any server-side session**. The use of sessions contradicts the ideas of HTTP because the communication is no longer stateless.

This condition makes it difficult to implement failover and load balancing. The only exception is in data that is required for authentication purposes in addition to the authentication information in the HTTP request.

## Browser controls

**The browser controls such as the back, forward, or refresh button should work**. This should be the norm, but many web applications with JavaScript logic have difficulties with it or have to do a lot of work in order to ensure the functioning of these buttons.

## No layout information in HTML

The **HTML may *not* contain layout information** and should be accessible via a screen reader. The layout is defined by CSS so that layout and content are separate.

## JavaScript only for enhancement

JavaScript can be used only in the form of *progressive enhancement*. The application can even be used without JavaScript – just not as easily and comfortably. The goal is not to avoid JavaScript completely, but to rely on the fundamental architecture and technologies of the web – HTTP, HTML, and CSS.

## No redundant logic

*Logic* must *not* be implemented redundantly on client and server because the business logic is implemented on the server and it shouldn't be implemented again on the client.

In the end, **ROCA applications are perfectly normal web applications**. They use web principles as they were originally intended.

QUIZ

## Quiz

1 Which of the following best describes ROCA?

COMPLETED 0%

1 of 3

In the next lesson, we'll discuss some benefits that this architecture provides.