[REST API security Stored token vs JWT vs OAuth](https://softwareengineering.stackexchange.com/questions/298973/rest-api-security-stored-token-vs-jwt-vs-oauth)

1. Token with HTTPS. I have used this approach a lot of times it is good enough with HTTPS. If user provides correct password and login, he will receive token in response, and will use it for the further requests. Token is generated by the server and stored, for instance in the table separate or the same where user info is stored. So for each request server checks if user has token and it is the same as in the database. Everything is pretty straightforward.
2. JWT Token. This token is self-descriptive, it contains all necessary information about the token itself, user cannot change for example expiration date or any other claim, because this token is generated (signed) by the server with secret keyword. This is also clear. But one big problem, personally for me, how to invalidate token.
3. OAuth 2. I don't understand why this approach should be used when communication is established directly between server and client. As far as I understand, OAuth server is used to issue token with restricted scope to allow other applications access user information without storing password and login. This is great solution for the social networks, when user wants to sign up on some page, server can request permissions to get user info, for instance from twitter or facebook, and fill registration fields with user data and so on.