API

WASA: Web and Software Architecture

Prof. Emanuele Panizzi

- definition of allowed interactions between two parts of software

WASA · API · Prof. Emanuele Panizzi · Sapienza University of Rome

2

- definition of allowed interactions between two parts of software
  - possible requests
  - their parameters
  - return values
  - any data format they require

CONSUMER <=> API <=> PROVIDER

WASA · API · Prof. Emanuele Panizzi · Sapienza University of Rome

3

- expilicit interface
- information hiding
- unbreakable contract

WASA · API · Prof. Emanuele Panizzi · Sapienza University of Rome

4

- APIs for programming languages
- operating system APIs
- software libraries
- hardware APIs
- web APIs

- private: e.g., for internal use of a company
- public: available for use by the public
    - access may be restricted to some users only. API tokens
- Interface stability: changes may break compatibility with clients
    - mark parts that may change because not stable yet (beta)
    - mark parts as 'deprecated' if they will be removed or not supported in the future

through documentation	through a description file
a free text explanation	*a machine-readable specification*
Possible cons:	Pros:
- incomplete or missing	- human- and machine-readable
- unclear	- automatic validation
- outdated	- code generation
	- mock servers (for testing)

- vendor-neutral description format for HTTP-based remote API
- industry standard for describing modern APIs
- broadly adopted

WASA · API · Prof. Emanuele Panizzi · Sapienza University of Rome

8

```yaml
openapi: 3.0.0
info:
  title: An example OpenAPI document
  description: |
    This API allows writing down marks on a Tic Tac Toe board
    and requesting the state of the board or of individual s
  version: 0.0.1
paths: {} # No endpoints defined yet
```

Hi-Lo Game

- I think a secret number between 1 and 100
- You try to guess it in max 10 trials
- Each trial, I will tell you if your guess is too high, or too low

WASA · API · Prof. Emanuele Panizzi · Sapienza University of Rome

10

1. start a new game
2. accept a guess (up to 10 guesses), and return hi, lo, or correct
3. reset a game, generating a new secret number
4. obtain the list of each guess in a game, with related results
5. obtain the list of all games, with final result (win/lose) and number of guesses

Which resources would you design?

WASA · API · Prof. Emanuele Panizzi · Sapienza University of Rome

11

- https://www.openapis.org
- https://oai.github.io/Documentation/