RESTest: Black-box constraint-based testing of RESTful web APIs

- **main idea**
  - generate test cases use inter-parameter dependencies
    - inter-parameter dependencies (or simply dependencies henceforth). For example, in the Google Maps API, when searching for places, if the location parameter is set, then the radius parameter must be set too, otherwise a 400 status code ("bad request") is returned

- **Inter-parameter Dependency Language (IDL) tool suite**
  - (IDL) is a domain-specific language for the specification of inter-parameter dependencies in web APIs. It
  - it provides support for eight different types of dependencies
    - Requires
    - ZeroOrOne
    - AllOrNone
    - Relational
    - Arithmetic
    - Complex
  - IDL4OSA is a extension for swagger that check inter parameter dependencies
  - IDLReasoner is tool for automated analysis of IDL specifications .

- **RESTest**
  - **take OAS as input**
    - if other format like RAML provided, convert them to OAS
    - optionaly we can use IDL4OAS for provide OAS dependencies
  - **generate test model** — it contanis OAS and some useful tunning
    - operation under test
    - authentication data
    - test data generators
    - weight
      - Testers might be interested in testing some parameters more thor- oughly than others, for example, those more used in practice. Weights allow to do so. A weight is a real number in the range [0,1]. The higher the weight of a parameter, the more frequently it will be used in test cases. By default, all parameters have a weight of 0.5.
  - **analysis test model** — using IDLReasoner that it get some useful fucntions
    - isValidRequest
      - This operation takes as input an API specification (including inter-parameter dependencies) and a service request (i.e., a list of parameters and their values), and returns a Boolean indicating whether the request is valid or not. A service request is valid if it satisfies all the inter-parameter dependencies defined in the specification
    - getRandomValidRequest — Subtopic
      - This operation receives the API specification of an API operation, and returns a random valid request for the operation, that is, a random assignment of values to input parameters satisfying all the dependencies of the specificatio
    - getRandomInvalidRequest. The
      - Contrary to the previous operation, this operation returns a random request violating one or more dependencies.
  - **generate abstract test cases**
    - Abstract test cases comprise test inputs, expected outputs (test oracles), and the required information to build the API request, e.g., the endpoint.
    - generates both nominal and faulty test cases
      - nominal test cases can be regarded as potentially valid test cases aimed at obtaining successful responses from the API (i.e., 2XX status codes)
      - Faulty test cases check the ability of the API to handle invalid inputs, and therefore they expect a client error as a response (i.e., 4XX status codes)
      - for generate faulty test cases use mutant
        - For example, RESTest supports the automated generation of faulty test cases by excluding mandatory parameters, using out- of-range values (e.g., assigning a string to an integer parameter), and violating the JSON schema of the request body, among others. Additionally,
    - RESTest supports the five test output
      - 5xx — server error
      - OAS — The response must conform to the OAS schema
      - 2xxp — If the request violates the specification of individual parameters (e.g., a mandatory parameter is missing), the status code must not be 2XX (suc- cessful response)
      - 2xxD — If the request violates one or more inter-parameter dependencies, the status code must not be 2XX
        - 2XXD and 4XX are novel contributions of RESTest
      - 4xx — If the request is valid according to the API specification, the status code must not be 4XX (client error response).
  - **generate executable tests**