

## Insurance App



You must implement the back-end system of an application used in an insurance company. The back end will expose REST endpoints with functionality, which are consumed by a separate front-end system. Also, it will store the data in an SQL database.

You must implement the following endpoints:

- **add an insurance policy** – allows a user to add a new policy in the database. The following business rules are defined for this endpoint:
  - If there is already a policy with the same policyNumber and type in the database, the request will be rejected, and the user will receive the message “There is already a policy with the same policy number and type”.
  - The request to add a policy has the following structure:

Field name	Present on the request	Mandatory on request	Present on response	Other validations
id	no	no	yes	PK, auto-generated on DB level
policyNumber	yes	yes	yes	String with max length of 30
type	yes	yes	yes	May be only one of the following values: house, vehicle, travel.
amountToPay	yes	yes	yes	Real positive value
amountInsured	yes	yes	yes	Real positive value
paymentTerms	yes	yes	yes	May be only one of the following values: once, quarterly, yearly, .
active	yes	yes	yes	Boolean value

- **get the list of all insurance policies** – allows a user to get a list with all the policies stored in the database. The user can optionally filter by the paymentTerms. The response will contain all the details of the policy (all the fields for each policy record).
- **delete an insurance policy** – allows a user to delete a policy stored in the database. The user must specify the id of the policy they want to delete. The deletion will be allowed only if the policy is inactive. The response will contain only a successful HTTP code, no body.

Unit tests should be implemented for the service methods, with a coverage of at least 70%.

The type of the SQL DB is chosen by you.