

# Dynamic risk type models

04/03/2021

Besart Kodraliu

## Overview

This is a simple implementation of dynamic risk type models in which the insurers can define their risk data models, adding an unlimited number of fields and different field types, and properties.

## Frontend Simple Demo (SPA VUE JS)

- **Add Risk Type Definition (/risk-type-add)**

Simple Vue Js view in which users can define their fields and properties.

Home | About | Risk Types | [Add Risk Type Definition](#)

### Add risk type definition

| Name              | Type    | Minimum | Maximum | Integer | Enum | Required |
|-------------------|---------|---------|---------|---------|------|----------|
| Vehicle Name      | string  |         | 20      |         |      | true     |
| Number of Airbags | integer |         | 5       |         |      | true     |

[Add Text Field](#) [Add Number Field](#) [Add Date Field](#) [Add Enum](#)

Field Name

Minimum

Maximum

Required

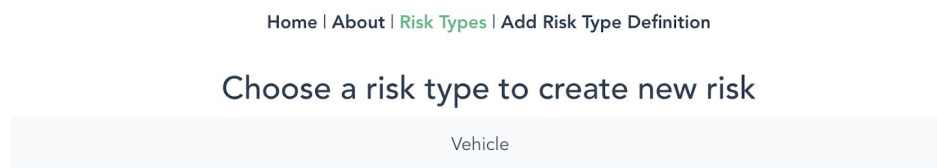
Add

☒

After defining fields and properties, users can write risk type name and save the form by clicking Submit Definition.

- **Risk Types (/risk-types)**

Simple Vue Js view in which users can see the risk type definitions, clicking one of them would send them in respective form.



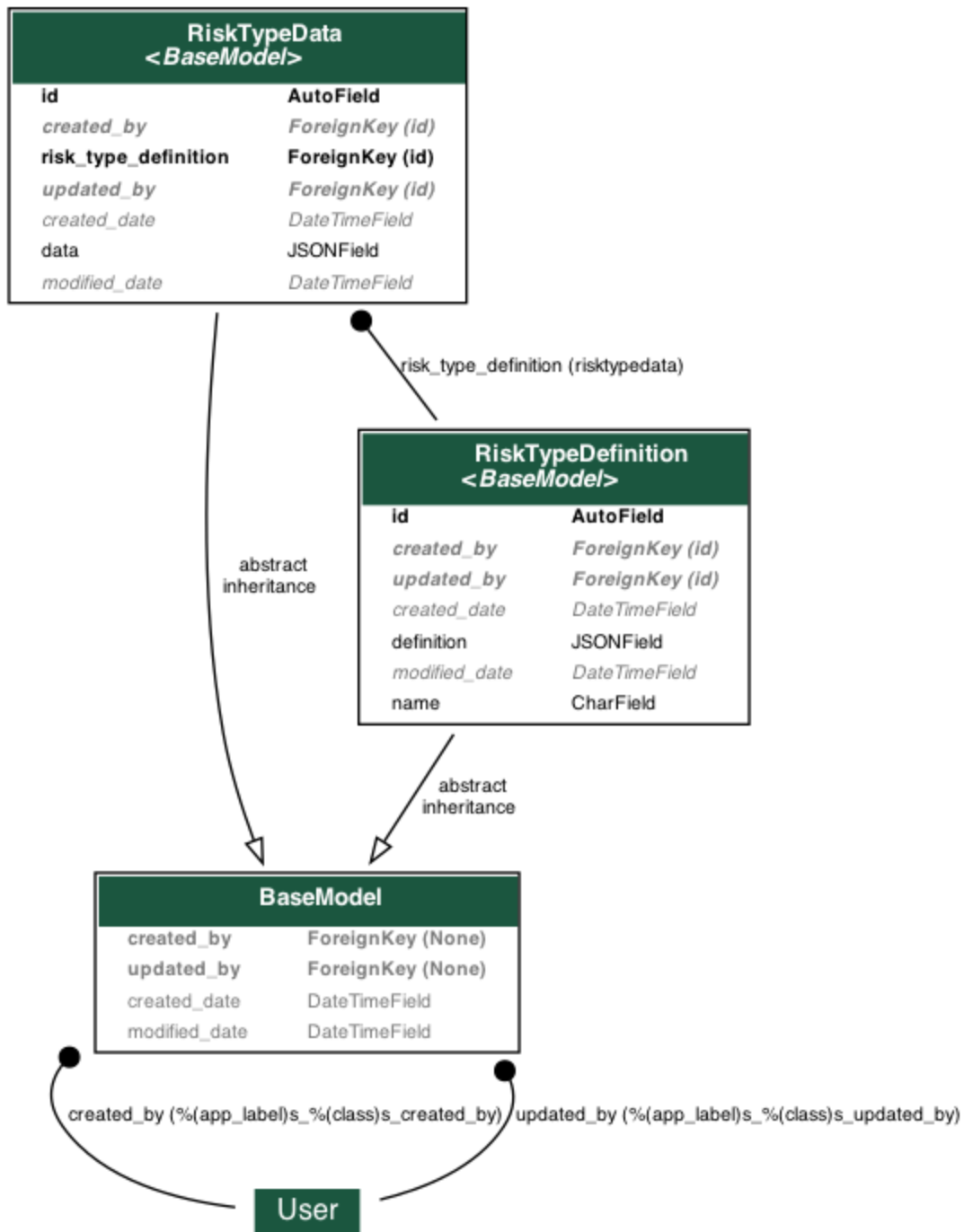
- **Risk Type Data Form (/risk-type/<risk\_type\_id> )**

Simple Vue Js view in which users can see generated form from the risk type definition, fields would have respective input type, and validations.

The screenshot shows a web application interface for the "Risk Type: Vehicle" form. At the top, there is a navigation bar with links: Home | About | Risk Types | Add Risk Type Definition. Below the navigation bar, the heading "Risk Type: Vehicle" is displayed. The form contains four input fields: "Vehicle Name" (text input), "Number of Airbags" (text input), "Purchase Date" (date input with a calendar icon), and "Color" (dropdown menu). At the bottom of the form, there is a green button labeled "Add Vehicle".

## Data Models

### UML Diagram



## ORM Classes

```
class BaseModel(models.Model):

    created_by = ForeignKey

    updated_by = ForeignKey

    created_date = DateTimeField

    modified_date = DateTimeField


class RiskTypeDefinition(BaseModel):

    id = AutoField

    name = CharField

    definition = JSONField


class RiskTypeData(BaseModel):

    id = AutoField

    risk_type_definition = ForeignKey to RiskTypeDefinition

    data = JSONField
```

## Backend (Django + Django Rest Framework)

### Endpoints

- Endpoint to get all risk type definitions

**GET** /risk-type-definitions/

### RESPONSE Example

```
[
  {
    "id": 37,
    "name": "Vehicle"
  }
]
```

```
]

```

- Endpoint to get single risk type definition

**GET** /risk-type-definitions/<id>

**RESPONSE Example**

```
{
  "id": "37"
  "name": "Vehicle"
  "definition": {valid Draft4 json schema}
}
```

Note: definition will generate form inputs and validations in frontend /risk-type/<id>

- Endpoint to create risk type definition

**POST** /risk-type-definitions/

```
{
  "name": "Vehicle"
  "definition": {valid Draft4 json schema}
}
```

Note: definition will be generated by frontend /risk-type-add form

and also will be validated in backend based on Draft4 json schema

- Endpoint to submit risk data

**POST** /risk-type-data/

```
{
  "risk_type_definition": "<id>"
  "data": {valid json schema, validated with risk_type_definition schema}
}
```