

## Equipo 8 - ciclo 4

### Integrantes:

Raúl Balaguera

María González

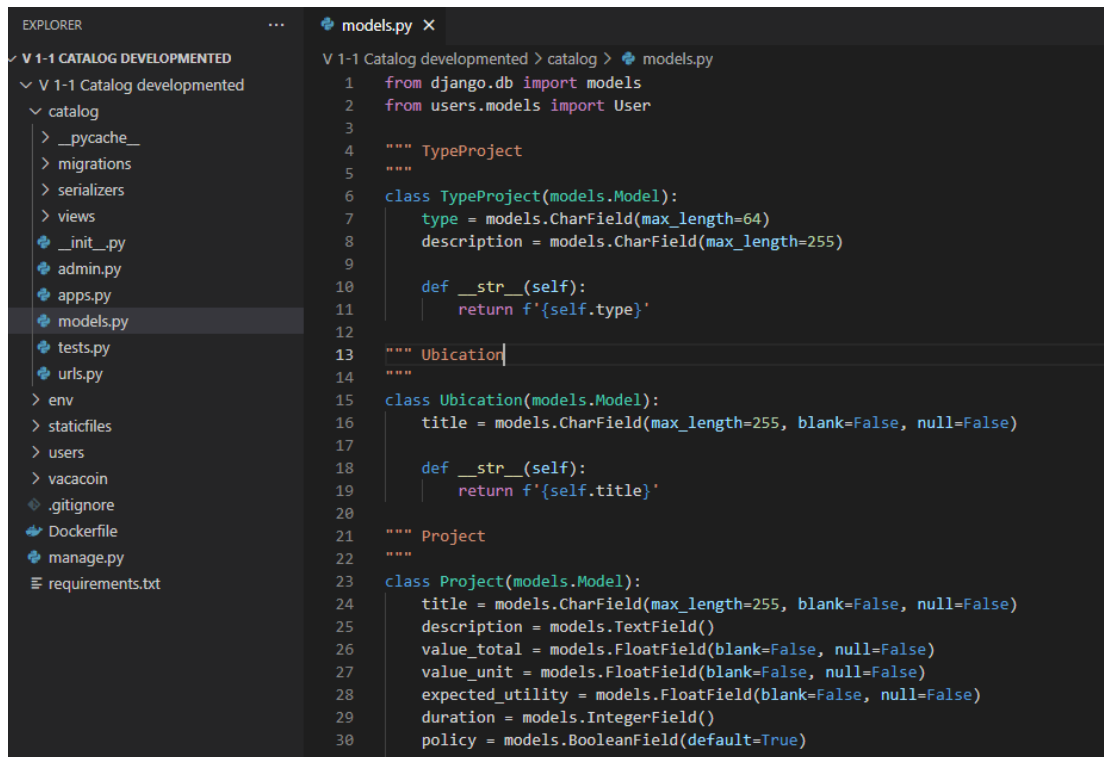
Santiago Jaramillo

Santiago Sanabria

Mauricio Murcia

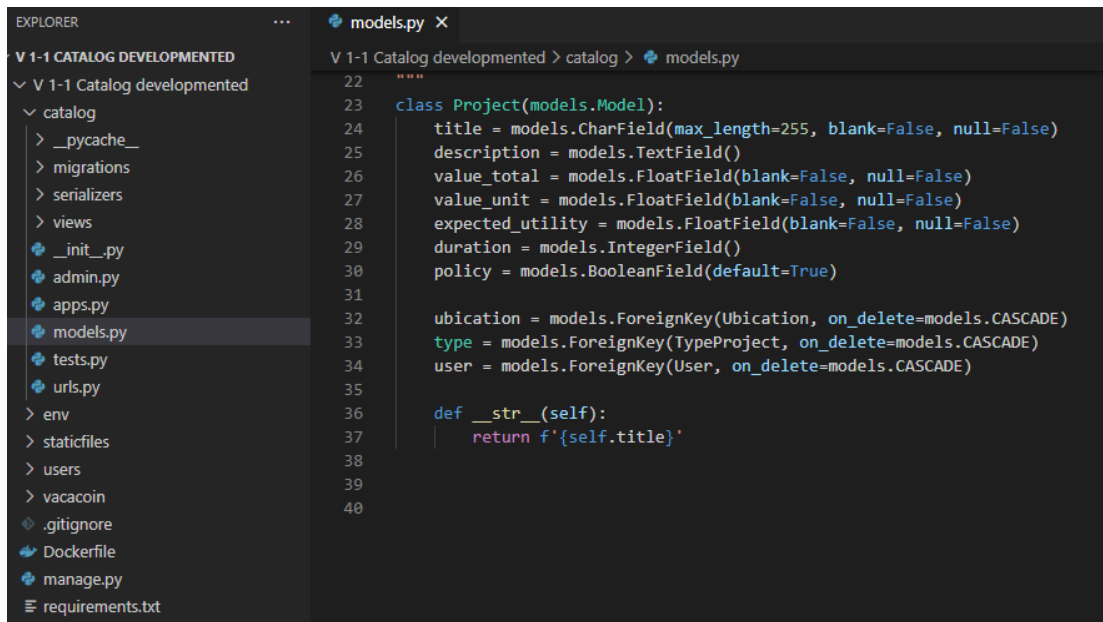
Daniel Pinto

## Models



The image shows a code editor with a dark theme. On the left is the 'EXPLORER' sidebar showing a project structure for 'V 1-1 CATALOG DEVELOPMENTED'. The 'models.py' file is selected and highlighted. The main editor area shows the content of 'models.py', which includes imports for Django models and a User object, and defines three model classes: TypeProject, Ubication, and Project.

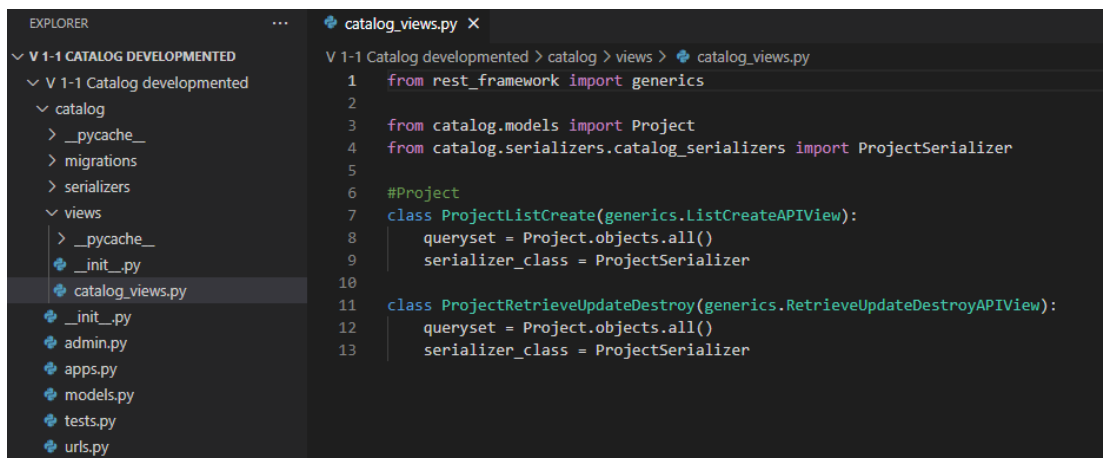
```
1 from django.db import models
2 from users.models import User
3
4 """ TypeProject
5 """
6 class TypeProject(models.Model):
7     type = models.CharField(max_length=64)
8     description = models.CharField(max_length=255)
9
10     def __str__(self):
11         return f'{self.type}'
12
13 """ Ubication
14 """
15 class Ubication(models.Model):
16     title = models.CharField(max_length=255, blank=False, null=False)
17
18     def __str__(self):
19         return f'{self.title}'
20
21 """ Project
22 """
23 class Project(models.Model):
24     title = models.CharField(max_length=255, blank=False, null=False)
25     description = models.TextField()
26     value_total = models.FloatField(blank=False, null=False)
27     value_unit = models.FloatField(blank=False, null=False)
28     expected_utility = models.FloatField(blank=False, null=False)
29     duration = models.IntegerField()
30     policy = models.BooleanField(default=True)
```



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows a project structure for 'V 1-1 CATALOG DEVELOPMENTED' with a 'catalog' folder containing various files. The 'models.py' file is selected. The Editor shows the code for the 'Project' model, which is a Django model with several fields and foreign key relationships.

```
22  """
23  class Project(models.Model):
24      title = models.CharField(max_length=255, blank=False, null=False)
25      description = models.TextField()
26      value_total = models.FloatField(blank=False, null=False)
27      value_unit = models.FloatField(blank=False, null=False)
28      expected_utility = models.FloatField(blank=False, null=False)
29      duration = models.IntegerField()
30      policy = models.BooleanField(default=True)
31
32      ubication = models.ForeignKey(Ubication, on_delete=models.CASCADE)
33      type = models.ForeignKey(TypeProject, on_delete=models.CASCADE)
34      user = models.ForeignKey(User, on_delete=models.CASCADE)
35
36      def __str__(self):
37          return f'{self.title}'
38
39
40
```

## Views



The screenshot shows the VS Code interface with the Explorer on the left and the Editor on the right. The Explorer shows the same project structure as the previous screenshot, but with the 'views' folder selected. The 'catalog\_views.py' file is selected. The Editor shows the code for the 'ProjectListCreate' and 'ProjectRetrieveUpdateDestroy' views, which are Django REST framework views.

```
1  from rest_framework import generics
2
3  from catalog.models import Project
4  from catalog.serializers.catalog_serializers import ProjectSerializer
5
6  #Project
7  class ProjectListCreate(generics.ListCreateAPIView):
8      queryset = Project.objects.all()
9      serializer_class = ProjectSerializer
10
11  class ProjectRetrieveUpdateDestroy(generics.RetrieveUpdateDestroyAPIView):
12      queryset = Project.objects.all()
13      serializer_class = ProjectSerializer
```

## Serializers

```
EXPLORER
V 1-1 CATALOG DEVELOPMENTED
  V 1-1 Catalog developmented
    catalog
      > __pycache__
      > migrations
      serializers
        > __pycache__
        > __init__.py
        catalog_serializers.py
      views
        > __pycache__
        > __init__.py
        catalog_views.py
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
    env
    staticfiles
    users
    vacacoin
    .gitignore
    Dockerfile
    manage.py
    requirements.txt

catalog_serializers.py X
V 1-1 Catalog developmented > catalog > serializers > catalog_serializers.py
1  from rest_framework import serializers
2
3  from catalog.models import Project, TypeProject, Ubication
4
5  """ TypeProjectSerializer
6  """
7  class TypeProjectSerializer(serializers.ModelSerializer):
8      class Meta:
9          model = TypeProject
10         fields = ['type']
11
12     """ UbicationSerializer
13     """
14     class UbicationSerializer(serializers.ModelSerializer):
15         class Meta:
16             model = Ubication
17             fields = '__all__'
18
19     """ ProjectSerializer
20     """
21     class ProjectSerializer(serializers.ModelSerializer):
22
23         """
24         ubication = UbicationSerializer(read_only=True)
25         type = TypeProjectSerializer()
26         """
27         class Meta:
28             model = Project
29             fields = ['id', 'title', 'description', 'value_total',
30                     'value_unit', 'expected_utility', 'duration',
31                     'policy', 'ubication', 'type', 'user']
32
33
```

## URLs

```
EXPLORER
V 1-1 CATALOG DEVELOPMENTED
  V 1-1 Catalog developmented
    catalog
      > __pycache__
      > migrations
      serializers
        > __pycache__
        > __init__.py
        catalog_serializers.py
      views
        > __pycache__
        > __init__.py
        catalog_views.py
        __init__.py
        admin.py
        apps.py
        models.py
        tests.py
        urls.py
    env
    staticfiles
    users
    vacacoin
    .gitignore
    Dockerfile
    manage.py
    requirements.txt

urls.py X
V 1-1 Catalog developmented > catalog > urls.py
1  from django.urls import path
2
3  from catalog.views.catalog_views import ProjectListCreate, ProjectRetrieveUpdateDestroy
4
5  urlpatterns = [
6      path('list-create/', ProjectListCreate.as_view()),
7      path('retrieve-update-destroy/<int:pk>', ProjectRetrieveUpdateDestroy.as_view())
8  ]
```

# Mongodb

ordenes.facturas

Documents

DOCUMENTS 4

TOTAL SIZE 1.3KB

AVG. SIZE 331B

INDEXES 1

TOTAL SIZE 36.9KB

AVG. SIZE 36.9KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

OPTIONS

FIND

RESET

...

ADD DATA

VIEW

...

Displaying documents 1 - 4 of 4

REFRESH

```
{
  "_id": ObjectId("61a1b274a59405cd079ed62d"),
  "cedula": "4589000",
  "telefono": "320958798",
  "logo": "vacacoin.png",
  "fecha": 2021-11-27T05:00:00.000+00:00,
  "productos": Array,
  "ciudad": "bogota",
  "direccioncliente": "diagonal 45 no 124-23",
  "nombrecliente": "pancho diaz",
  "valortotal": "320000"
}
```

```
{
  "_id": ObjectId("61a419409b20ac5586075dc2"),
  "cedula": "1.567.435.789",
  "telefono": "3157438765",
  "logo": "vacacoin.png",
  "fecha": 2021-11-27T05:00:00.000+00:00,
  "productos": Array,
  "ciudad": "zipaquira",
  "direccioncliente": "diagonal 45 no 124-23",
  "nombrecliente": "Poncio Pilatos",
  "valortotal": "3.567.432"
}
```

ordenes.lotesxtransacción

Documents

DOCUMENTS 5

TOTAL SIZE 1.0KB

AVG. SIZE 204B

INDEXES 1

TOTAL SIZE 36.9KB

AVG. SIZE 36.9KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

OPTIONS

FIND

RESET

...

ADD DATA

VIEW

...

Displaying documents 1 - 5 of 5

REFRESH

```
{
  "_id": ObjectId("61a41b769b20ac5586075dd4"),
  "tipo": "vacas para engorde",
  "cantidad": 20,
  "valor unitario": 5000,
  "descripcion": "vacas tipo brangus para ceba intensiva con ",
  "impuesto": 0.18,
  "ganadero": "Marbelle"
}
```

```
{
  "_id": ObjectId("61a41be79b20ac5586075ddd"),
  "tipo": "vacas para engorde",
  "cantidad": 100,
  "valor unitario": "Valor del Tesoro Nacional",
  "descripcion": "vacas tipo brangus para ceba intensiva con ",
  "impuesto": 0.18,
  "ganadero": "Rastacuando"
}
```

```
{
  "_id": ObjectId("61a41c6e9b20ac5586075de6"),
  "tipo": "vacas para engorde",
  "cantidad": 850,
  "valor unitario": "1000000",
  "descripcion": "vacas tipo brangus para ceba intensiva con ",
  "impuesto": 0.18,
  "ganadero": "Alvaro Uribe Velez"
}
```



ordenes.facturas Documents

DOCUMENTS 4 TOTAL SIZE 1.3KB AVG. SIZE 331B INDEXES 1 TOTAL SIZE 36.9KB AVG. SIZE 36.9KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA VIEW ...

Displaying documents 1 - 4 of 4 REFRESH

```

valortotal: "3.567.432"

{
  "_id": ObjectId("61a419be9b20ac5586075dc3"),
  "cedula": "9.765.432",
  "telefono": "3164359087",
  "logo": "vacacoin.png",
  "fecha": 2021-11-27T05:00:00.000+00:00,
  "productos": Array,
  "ciudad": "Bogotá",
  "direccioncliente": "diagonal 4 no 134 - 70",
  "nombrecliente": "Ivan Duque",
  "valortotal": "40.786.875"
}

{
  "_id": ObjectId("61a41a3c9b20ac5586075dc4"),
  "cedula": "54.678.543",
  "telefono": "311346785",
  "logo": "vacacoin.png",
  "fecha": 2021-11-27T05:00:00.000+00:00,
  "productos": Array,
  "ciudad": "Cajica",
  "direccioncliente": "Carrera 7 N 29 34",
  "nombrecliente": "Gustavo Petro",
  "valortotal": "1.456.345.899"
}

```

## Conexión db

IntelliJ - Apache NetBeans IDE 12.0

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl-F)

Projects: Files Services

- Source Packages
  - com.mycompany.jmongo
    - conexion.java
- Test Packages
- Dependencies
  - mongo-4.0.6.jar
  - mongodb-driver-sync-4.0.6.jar
  - mongodb-driver-core-4.0.6.jar
- Java Dependencies
- Project Files
  - pom.xml

Navigation: POM model

- Model Version: 4.0.0
- GroupId: com.mycompany
- ArtifactId: jmongo
- Packaging: jar
- Version: 1.0-SNAPSHOT
- Dependencies
- Properties

Start Page pom.xml [Mongo] conexon.java

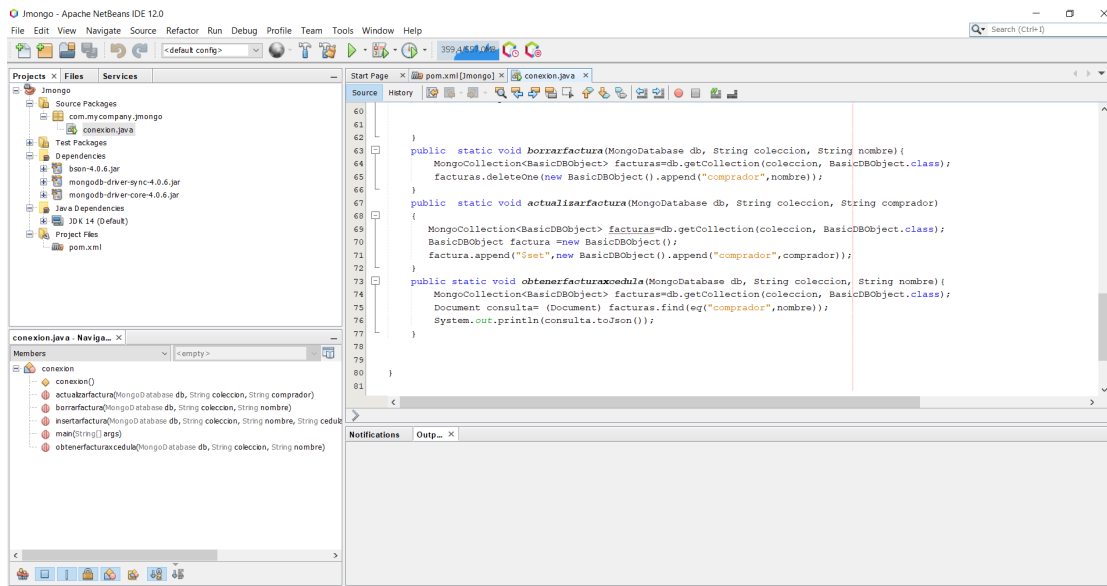
```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany</groupId>
  <artifactId>jmongo</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.mongodb</groupId>
      <artifactId>mongodb-driver-sync</artifactId>
      <version>4.0.6</version>
    </dependency>
    <dependency>
      <groupId>org.mongodb</groupId>
      <artifactId>bson</artifactId>
      <version>4.0.6</version>
    </dependency>
  </dependencies>
</project>

```

Notifications Outp...

1:1 | BNS: UNK (LF)



```

}

public static void borrarfactura(MongoDatabase db, String coleccion, String nombre){
    MongoCollection<BasicDBObject> facturas=db.getCollection(coleccion, BasicDBObject.class);
    facturas.deleteOne(new BasicDBObject().append("comprador",nombre));
}

public static void actualizarfactura(MongoDatabase db, String coleccion, String comprador)
{
    MongoCollection<BasicDBObject> facturas=db.getCollection(coleccion, BasicDBObject.class);
    BasicDBObject factura =new BasicDBObject();
    factura.append("$set",new BasicDBObject().append("comprador",comprador));
}

public static void obtenerfacturaxcedula(MongoDatabase db, String coleccion, String nombre){
    MongoCollection<BasicDBObject> facturas=db.getCollection(coleccion, BasicDBObject.class);
    Document consulta= (Document) facturas.find(eq("comprador",nombre));
    System.out.println(consulta.toJson());
}

}

```

```

public class conexion {
    public static void main(String[]args){

        MongoClient client = MongoClient.create("mongodb+srv://VCAdmin:Vacacoin3102@dbvc.aywz1.mongodb.net/myFirstDatabase?retryWrites=true&w=majority");
        MongoDatabase database = client.getDatabase("ordenes");
        System.out.println("entre");
        MongoCollection<BasicDBObject> facturas = database.getCollection("facturas", BasicDBObject.class);
        System.out.println("obtuve");
        BasicDBObject documento=new BasicDBObject();
        documento.put("nombre", "pancho juarez");
        facturas.insertOne(documento);
        //insertarfactura(database,"facturas","juan roso","112345621",24,4000000,0.18,"2020-11-2019");

    }

    public static void insertarfactura(MongoDatabase db, String coleccion,String nombre,String cedula,int cantidad,
    MongoCollection<BasicDBObject> facturas = db.getCollection(coleccion, BasicDBObject.class);
    BasicDBObject documento=new BasicDBObject();
    documento.put("comprador",nombre);
    documento.put("cedula", cedula);
    documento.put("fecha", fecha);
    documento.put("total",total);
}

```

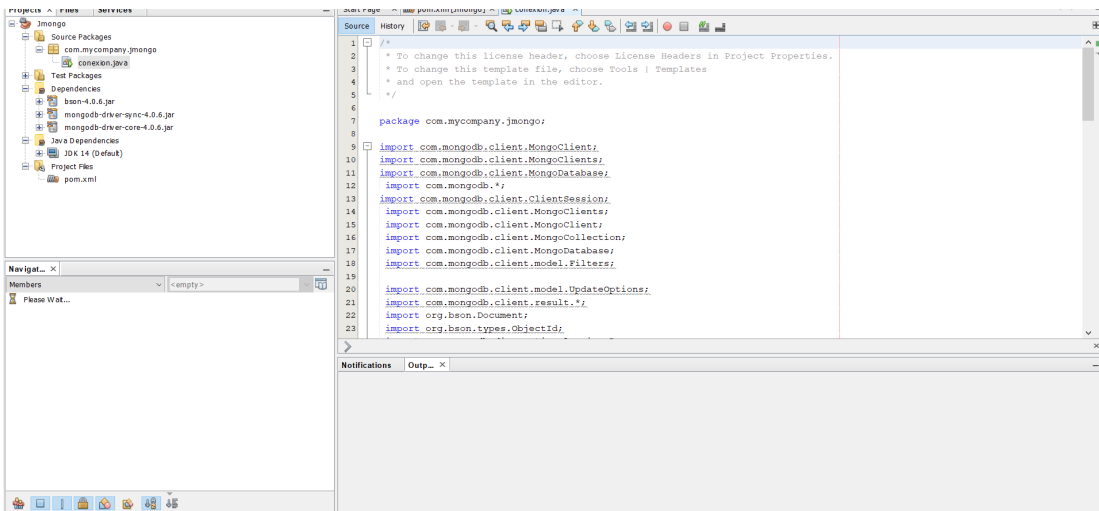
```

import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;

import com.mongodb.client.model.UpdateOptions;
import com.mongodb.client.result.*;
import org.bson.Document;
import org.bson.types.ObjectId;
import com.mongodb.diagnostics.logging.Loggers;

import java.util.List;
import java.util.Arrays;
import java.util.ArrayList;
import java.util.logging.*;
import static com.mongodb.client.model.Filters.*;
import java.util.Date;

```





## Links

**Drive:**

<https://drive.google.com/drive/folders/1HnCPo83hi2uDs5VZ6AhZRcm0jIDIVUFQ>

**Repo:**

<https://github.com/Ravercrown/4a-docs/>