

Specifiche di progetto

- Visionare il data-set di esempio disponibile all'indirizzo consegnato al gruppo in fase di assegnazione del progetto contenuto nella email inviata dai docenti
- Creare un insieme di classi che modellino il data-set sfruttando i principi dell'OOP (incapsulamento, ereditarietà, polimorfismo, classi astratte, interfacce...)
- Progettare (mediante UML: casi d'uso, diagramma delle classi e diagrammi delle sequenze) e sviluppare in JAVA una applicazione che implementi le seguenti funzionalità:
 - **all'avvio:** effettui il download del data-set (zip o csv) che contiene dati in formati CSV partendo dall'indirizzo fornito dopo opportuna decodifica del JSON che contiene la URL utile per scaricare il file (si veda esempio più avanti); si può prevedere che tale fase venga eseguita al primo avvio dell'applicazione;
 - **al download completato:** effettui il parsing dei dati creando delle strutture dati opportune sulla base delle classi di cui sopra (ogni record del data-set corrisponde ad un oggetto di una classe); si presti attenzione al parsing del CSV dove in alcuni casi vi potrebbero essere più caratteri separatori che devono essere opportunamente gestiti.
 - **su richiesta** mediante API REST GET o POST (per evitare query string complesse) con rotte distinte **usando Spring Boot:**
 - restituire i metadati (formato JSON) ovvero elenco degli attributi e del tipo
 - restituire i dati (formato JSON);
 - restituire statistiche sui dati (formato JSON) che specificando l'attributo sul quale effettuare la computazione (colonna dei dati) quali:
 - Numeri:
 - avg, min, max, dev std, sum, count
 - Stringhe
 - Conteggio elementi unici (per ogni elemento unico indicare il numero di occorrenze)
 - la restituzione deve prevedere la possibilità di specificare in fase di richiesta una serie di filtri su attributi con operatori *conditional* e *logical* (*almeno 2 operatori conditional e 2 logical*)
 - Gestire problemi durante il processo di importazione (es. numero di attributi errato in una data riga)
 - Gestire problemi nella fase di richiesta (es. si richiedono statistiche per un attributo inesistente oppure i filtri non sono corretti)
- Documentare il codice sviluppato (attraverso commenti esaustivi e l'utilizzo di JavaDoc)
- Usare github per versionare il codice da consegnare
- Usare il README.md per creare una pagina descrittiva del progetto inserendo i diagrammi UML e quanto necessario alla comprensione del funzionamento del vostro programma
 - modalità implementative delle statistiche e dei filtri
 - elencare esempi di test che saranno utilizzati per la verifica delle funzionalità
- Si raccomanda una organizzazione del codice in package

In learn vi sono vari esempi che possono essere di aiuto per lo svolgimento del progetto

Esempi di Filtri (si dà al gruppo la possibilità di effettuare personalizzazione della sintassi degli operatori di seguito presentanti)

Logical operators overview

Operator	Description	Example
\$not	Negation logical operator	<code>{"field" : {"\$not" : val}}</code>
\$in	Match any value in array	<code>{"field" : {"\$in" : [value1, value2, ...]}}</code>
\$nin	Not match any value in array	<code>{"field" : {"\$nin" : [value1, value2, ...]}}</code>
\$or	Logical operator	<code>{"\$or": [{"status": "GOLD"}, {"status": "SILVER"}]}</code>
\$and	Logical operator	<code>{"\$and": [{"status": "GOLD"}, {"sales": 1000}]}</code>

Conditional operators

Operator	Description	Example
\$gt	>	<code>{"salary": {"\$gt": 10000}}</code>
\$gte	>=	<code>{"salary": {"\$gte": 10000}}</code>
\$lt	<	<code>{"salary": {"\$lt": 10000}}</code>
\$lte	<=	<code>{"salary": {"\$lte": 10000}}</code>
\$bt	>= value <=	<code>{"salary": {"\$bt": [5000, 7500]}}</code>

Esempio:

Esempio di URL condiviso dai docenti:

http://data.europa.eu/euodp/data/api/3/action/package_show?id=eu-wide-stress-test-results-2018-asset-quality

Da ciò si ottiene il seguente JSON con il quale effettuare il download. Si faccia riferimento al CSV (in particolare il la chiave format che conterrà la stringa <http://publications.europa.eu/resource/authority/file-type/CSV> che identifica un file di tipo CSV).

```
{
  "help": "Return the metadata of a dataset (package) and its resources.\n  This overrides core package_show to deal with DCAT-AP data\n\n  :param str uri: the uri of the dataset\n\n  :rtype: dictionary\n\n  ",
  "success": true,
  "result": {
    "maintainer": null,
    "private": false,
    "maintainer_email": null,
    "revision_timestamp": null,
    "keywords": [
      {
        "display_name": "capital ratio",
        "name": "capital ratio"
      },
      {
        "display_name": "banks",
        "name": "banks"
      },
      {
        "display_name": "stress tests",
        "name": "stress tests"
      }
    ],
    "metadata_modified": "2019-04-09T14:34:04",
    "temporal_granularity": "",
    "concepts_eurovoc": [
      "http://eurovoc.europa.eu/4738",
      "http://eurovoc.europa.eu/427"
    ],
    "license_id": null,
    "type": "dataset",
    "resources": [
      {

```

```

        "mimetype": null,
        "mimetype_inner": "",
        "hash": "",
        "description": "This Excel file allows to visualise data reported in the templates: non-performing exposures,
forborne exposures and collateralised loans. Figures are aggregated by country of the banks.",
        "format": "http://publications.europa.eu/resource/authority/file-type/XLS",
        "url": "https://eba.europa.eu/documents/10180/2419197/Credit%20Risk_NPE_Forborne%20by%20Country_2018.xlsxb",
        "created": "2018-12-21 13:53:22.069860",
        "tracking_summary": null,
        "name": "NPE forborne and collateral - Data aggregated by countries of banks",
        "state": "http://publications.europa.eu/resource/authority/dataset-status/COMPLETED",
        "last_modified": "",
        "download_total_resource": "5",
        "url_type": null,
        "position": null,
        "iframe_code": "",
        "datastore_active": false,
        "id": "http://data.europa.eu/88u/distribution/3b9cda25-a87d-4717-bdf2-1953082b2d95",
        "resource_type": "http://publications.europa.eu/resource/authority/distribution-type/FEED_INFO",
        "size": ""
    },
    {
        "mimetype": null,
        "mimetype_inner": "",
        "hash": "",
        "description": "This Excel file provides country aggregated credit risk exposures for a specific country of
counterparty (US, DE, ...) broken down by the country of the banks exposed to it (AT, DE,..) for all regulatory portfolios (IRB/SA)
and exposure classes (corporates, retail, ...).",
        "format": "http://publications.europa.eu/resource/authority/file-type/XLS",
        "url": "https://eba.europa.eu/documents/10180/2419197/Credit%20Risk_Aggregates%20by%20Counterparty_2018.xlsxb",
        "created": "2018-12-21 13:53:22.069875",
        "tracking_summary": null,
        "name": "Credit Risk (STA / IRB) - Data aggregated by countries of counterparties",
        "state": "http://publications.europa.eu/resource/authority/dataset-status/COMPLETED",
        "last_modified": "",
        "download_total_resource": "17",
        "url_type": null,
        "position": null,
        "iframe_code": "",
        "datastore_active": false,
        "id": "http://data.europa.eu/88u/distribution/0a7892b0-05c3-4b60-84f2-cbc4e59df69d",
        "resource_type": "http://publications.europa.eu/resource/authority/distribution-type/FEED_INFO",
        "size": ""
    },
    {
        "mimetype": null,
        "mimetype_inner": "",
        "hash": "",
        "description": "This Excel file provides country aggregated credit risk exposures for a specific country (AT, DE,
...) broken down by regulatory portfolio (IRB/SA) and exposure class (corporates, retail, ...), towards all the countries of the
counterparty.",
        "format": "http://publications.europa.eu/resource/authority/file-type/XLS",
        "url": "https://eba.europa.eu/documents/10180/2419197/Credit%20Risk_Aggregates%20by%20Country_2018.xlsxb",
        "created": "2018-12-21 13:53:22.069870",
        "tracking_summary": null,
        "name": "Credit Risk (STA / IRB) - Data aggregated by countries of banks",
        "state": "http://publications.europa.eu/resource/authority/dataset-status/COMPLETED",
        "last_modified": "",
        "download_total_resource": "10",
        "url_type": null,
        "position": null,
        "iframe_code": "",
        "datastore_active": false,
        "id": "http://data.europa.eu/88u/distribution/ab3c8247-fd62-42d5-b3d1-3ee4947161a1",
        "resource_type": "http://publications.europa.eu/resource/authority/distribution-type/FEED_INFO",
        "size": ""
    },
    {
        "mimetype": null,
        "mimetype_inner": "",
        "hash": "",
        "description": "Credit risk raw data",
        "format": "http://publications.europa.eu/resource/authority/file-type/CSV",
        "url": "https://eba.europa.eu/documents/10180/2419191/TRA_CR.csv",
        "created": "2018-12-21 14:04:29.709875",
        "tracking_summary": null,
        "name": "Credit risk raw data",
        "state": "http://publications.europa.eu/resource/authority/dataset-status/COMPLETED",
        "last_modified": "",
        "download_total_resource": "36",
        "url_type": null,
        "position": null,
        "iframe_code": "",
        "datastore_active": false,
        "id": "http://data.europa.eu/88u/distribution/4eff0ed7-89d1-4010-a400-7b865calfbe5",
        "resource_type": "http://publications.europa.eu/resource/authority/distribution-type/FEED_INFO",
        "size": ""
    },
    ...
}

```

Le chiamate di seguito sono solo a titolo di esempio

Esempio di GET (in riferimento ad un dataset disponibile presso <https://www.istat.it/storage/codici-unita-amministrative/Elenco-comuni-italiani.xls>)

```

GET /metadata/ (ottiene i metadati)
[
{
    "alias": "CodReg",
    "sourceField": "Codice Regione",

```

```

    "type": "string"
  },
  {
    "alias": "CodUTS",
    "sourceField": "Codice dell'Unità territoriale sovracomunale \n(valida a fini statistici)",
    "type": "string"
  },
  {
    "alias": "CodProv",
    "sourceField": "Codice Provincia (Storico) (1)",
    "type": "string"
  },
  {
    "alias": "Progressivo",
    "sourceField": "Progressivo del Comune (2)",
    "type": "string"
  },
  {
    "alias": "CodComAlfa",
    "sourceField": "Codice Comune formato alfanumerico",
    "type": "string"
  },
  {
    "alias": "Den",
    "sourceField": "Denominazione (Italiana e straniera)",
    "type": "string"
  },
  {
    "alias": "DenITA",
    "sourceField": "Denominazione in italiano",
    "type": "string"
  },
  {
    "alias": "",
    "sourceField": "Denominazione altra lingua",
    "type": "string"
  },
  {
    "alias": "CodRipGeo",
    "sourceField": "Codice Ripartizione Geografica",
    "type": "string"
  },
  {
    "alias": "RipGeo",
    "sourceField": "Ripartizione geografica",
    "type": "string"
  },
  {
    "alias": "DenReg",
    "sourceField": "Denominazione regione",
    "type": "string"
  },
  {
    "alias": "DenUTS",
    "sourceField": "Denominazione dell'Unità territoriale sovracomunale \n(valida a fini
statistici)",
    "type": "string"
  },
  {
    "alias": "Flag",
    "sourceField": "Flag Comune capoluogo di provincia/città metropolitana/libero consorzio",
    "type": "integer"
  },
  {
    "alias": "SiglaAuto",
    "sourceField": "Sigla automobilistica",
    "type": "string"
  },
  {
    "alias": "CodComNum",
    "sourceField": "Codice Comune formato numerico",
    "type": "integer"
  },
  {
    "alias": "CodComNum110",
    "sourceField": "Codice Comune numerico con 110 province (dal 2010 al 2016)",
    "type": "integer"
  },

```

```

{
  "alias": "CodComNum107",
  "sourceField": "Codice Comune numerico con 107 province (dal 2006 al 2009)",
  "type": "integer"
},
{
  "alias": "CodComNum103",
  "sourceField": "Codice Comune numerico con 103 province (dal 1995 al 2005)",
  "type": "integer"
},
{
  "alias": "CodCatastale",
  "sourceField": "Codice Catastale del comune",
  "type": "string"
},
{
  "alias": "Popo2011",
  "sourceField": "Popolazione legale 2011 (09/10/2011)",
  "type": "integer"
},
{
  "alias": "NUTS1",
  "sourceField": "NUTS1",
  "type": "string"
},
{
  "alias": "NUTS2",
  "sourceField": "NUTS2(3)",
  "type": "string"
},
{
  "alias": "NUTS3",
  "sourceField": "NUTS3",
  "type": "string"
}
}
]

```

GET /data (tutti i dati)

GET /data?filter={"Popo2011": {"\$bt": [1000, 10000]}} (solo quelli tra 1000 e 10000)

In alternativa i parametri possono essere passati mediante una POST (nel body della richiesta)

```

[
  {
    "CodReg": 11,
    "CodUTS": "042",
    "CodProv": "042",
    "Progressivo": "002",
    "CodComAlfa": "042002",
    "Den": "Ancona",
    "DenITA": "Ancona",
    "": "",
    "CodRipGeo": 3,
    "RipGeo": "Centro",
    "DenReg": "Marche",
    "DenUTS": "Ancona",
    "Flag": 1,
    "SiglaAuto": "AN",
    "CodComNum": 42002,
    "CodComNum110": 42002,
    "CodComNum107": 42002,
    "CodComNum103": 42002,
    "CodCatastale": "A271",
    "Popo2011": 100.497,
    "NUTS1": "ITI",
    "NUTS2": "ITI3",
    "NUTS3": "ITI32"
  },
  {
    "CodReg": 11,
    "CodUTS": "042",
    "CodProv": "042",
    "Progressivo": "003",
    "CodComAlfa": "042003",
    "Den": "Arcevia",
    "DenITA": "Arcevia",
    "": ""
  }
]

```

```

    "CodRipGeo": 3,
    "RipGeo": "Centro",
    "DenReg": "Marche",
    "DenUTS": "Ancona",
    "Flag": 0,
    "SiglaAuto": "AN",
    "CodComNum": 42003,
    "CodComNum110": 42003,
    "CodComNum107": 42003,
    "CodComNum103": 42003,
    "CodCatastale": "A366",
    "Popo2011": 4.914,
    "NUTS1": "ITI",
    "NUTS2": "ITI3",
    "NUTS3": "ITI32"
  }
...
]
GET /stats?field=Popo2011

[
  {
    "field": "Popo2011",
    "avg": 7509,
    "min": 30,
    "max": 2617175,
    "std": 40177,
    "sum": 59433744,
    "count": 7915
  }
]
GET /stats?field=Popo2011&filter={"Popo2011": {"$bt": [1000, 10000]}}

[
  {
    "field": "Popo2011",
    "avg": 3610,
    "min": 1001,
    "max": 10000,
    "std": 2300,
    "sum": 17522288,
    "count": 4855
  }
]

```