/03

SOFTWARE

Crafted UI & Responsive Design

Crafted and fine-tuned UI

Consistent Look

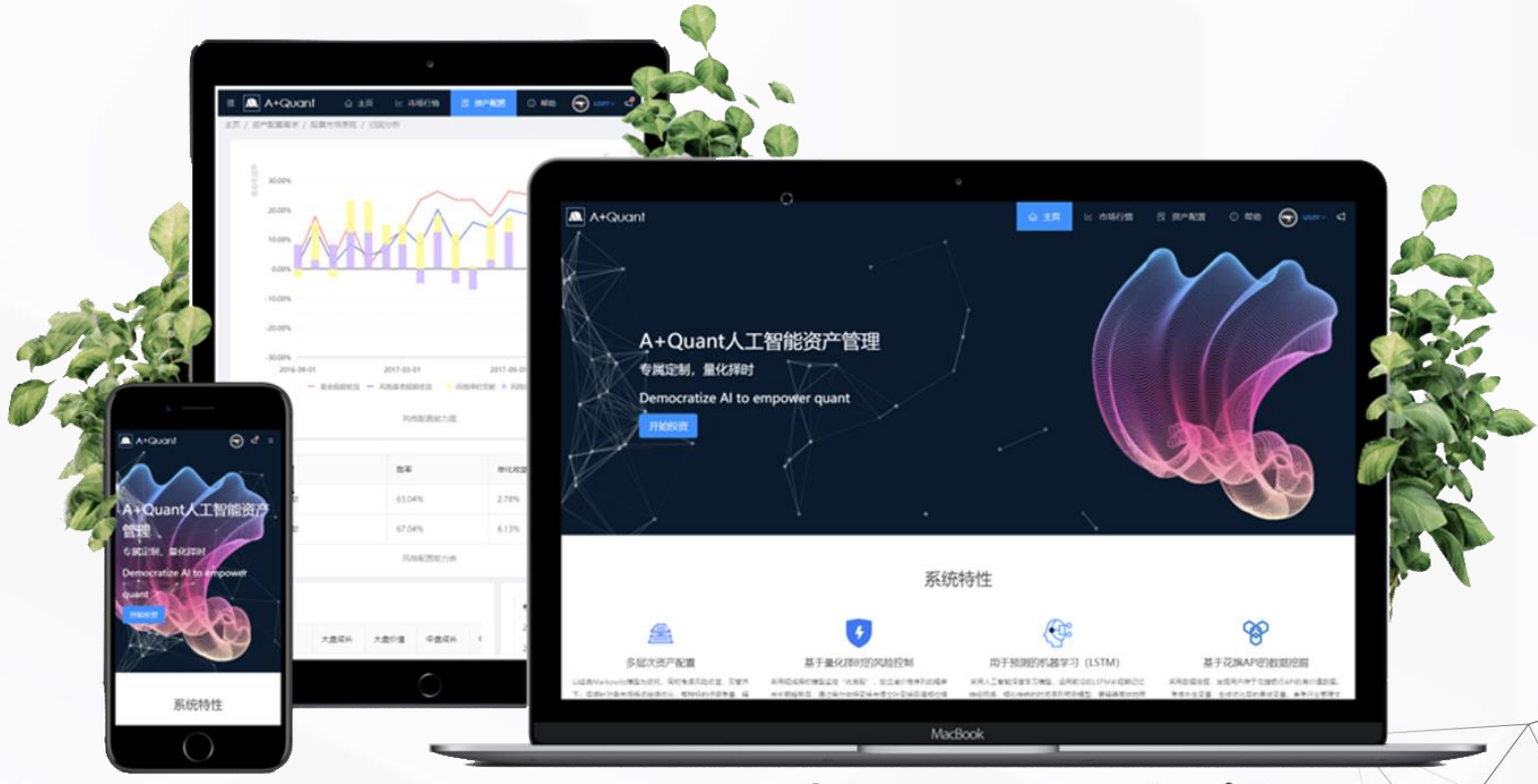
Fluent animations

Rich and interactive charts

Responsive design

Optimized for Mobile, Tablet & PC

With Ant Design, AntV and Bizcharts



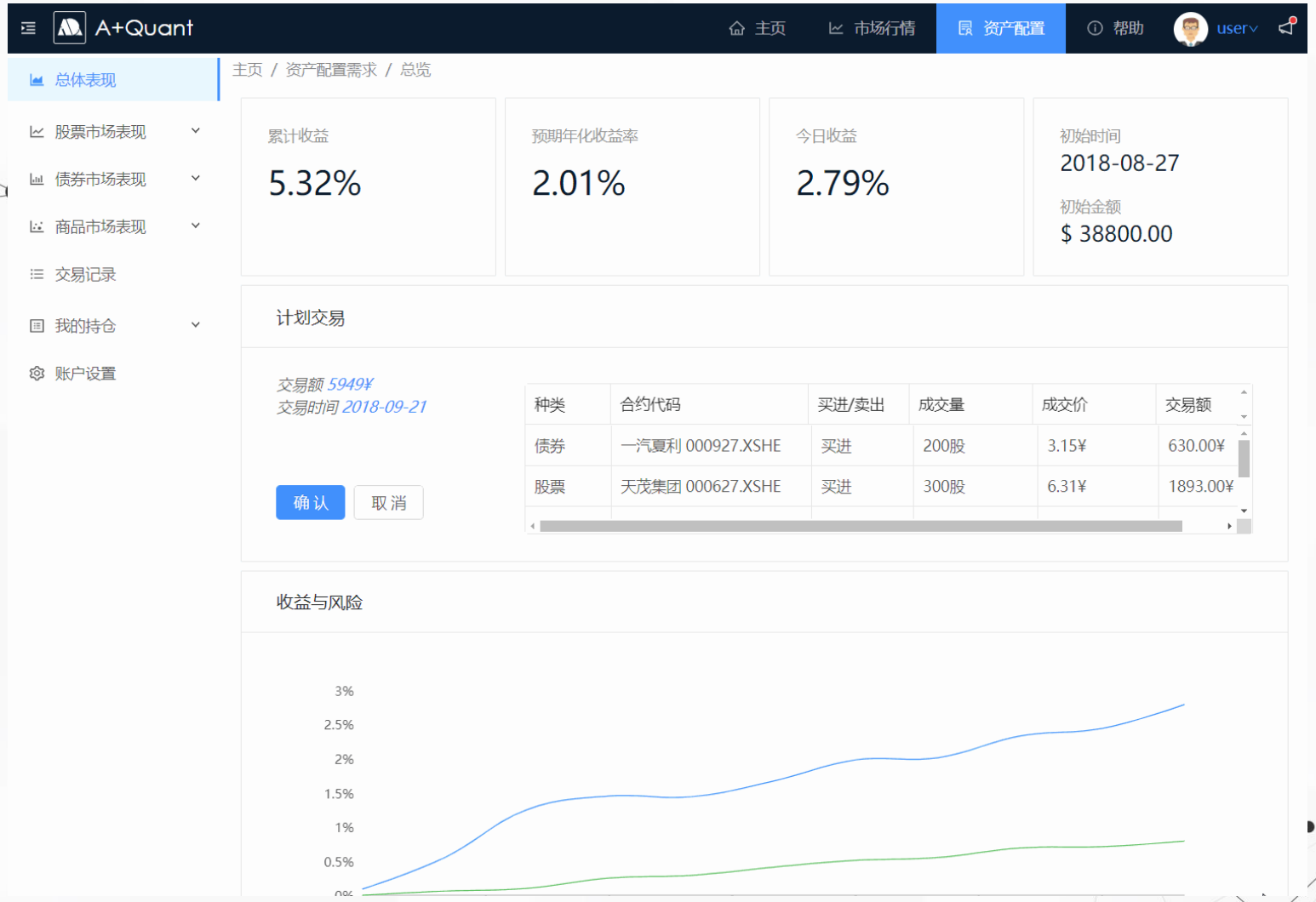
Crafted UI & Responsive Design

UI | Architecture | Workflow

Homepage

Crafted UI & Responsive Design

UI | Architecture | Workflow



Rich and interactive charts

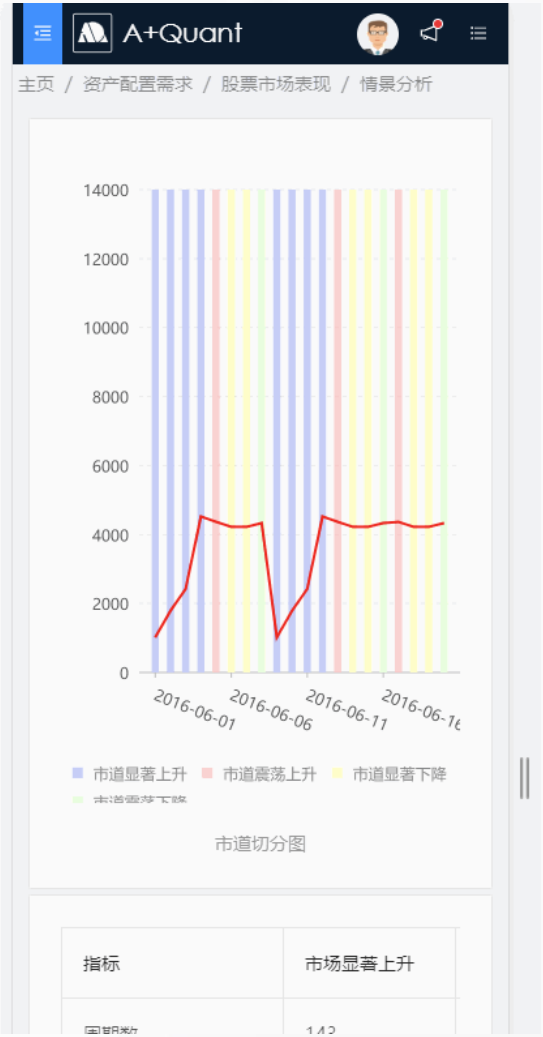
Crafted UI & Responsive Design

UI | Architecture | Workflow

Mobile Friendly

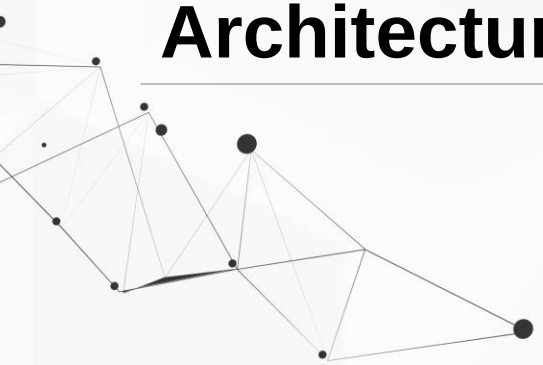
Crafted UI & Responsive Design

UI | Architecture | Workflow

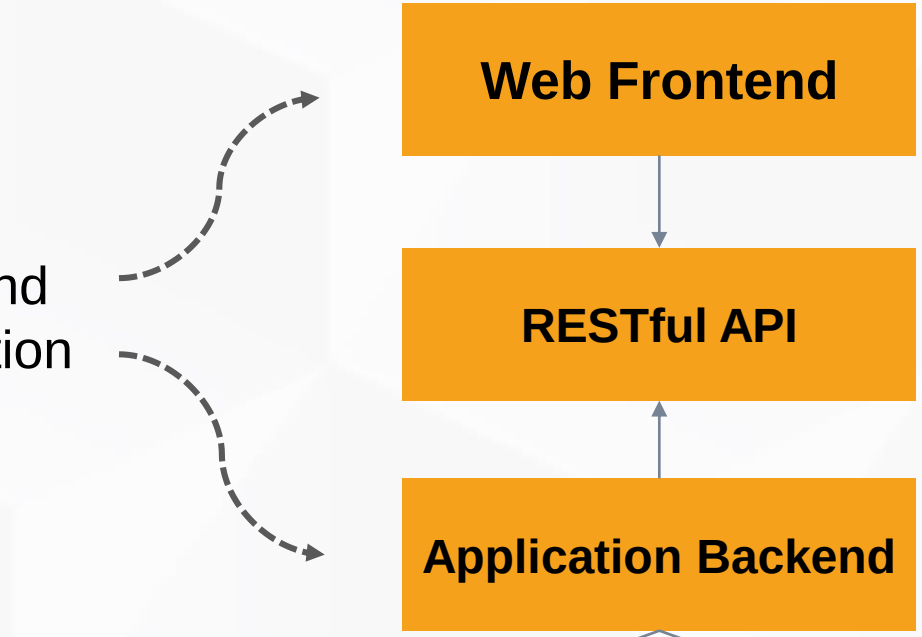


Responsive Design

Architecture



Complete Frontend
Backend Separation



Improve network performance

Easier integration of other clients

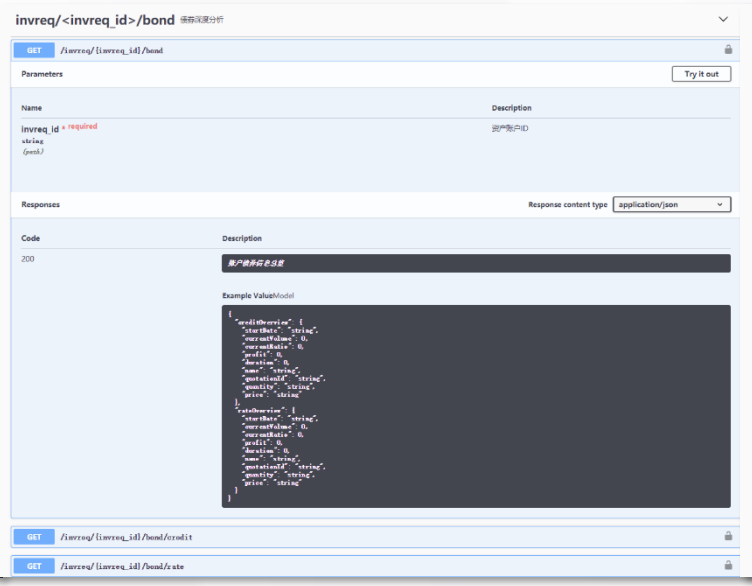
Complete Frontend Backend Separation

RESTful API with *Swagger*

Auto Generate

http://118.25.55.247:5000/api/

```python
bond_overview_data = ns.model("债券总体数据", {
    'startDate': fields.String(description="初始日期，格式yy-mm-dd"),
    'currentVolume': fields.Float(description="当前金额"),
    'currentRatio': fields.Float(description="当前债券占整个配置的百分比:百分数"),
    'profit': fields.Float(description="收益 ？金额or百分数"),  # Y
    'duration': fields.Float(description="久期"),  # N
    'name': fields.String(description="债券标的名字"),  # Y
    'quotationId': fields.String(description="债券标的ID"),  # Y
    'quantity': fields.String(description="数量"),  # Y
    'price': fields.String(description="单价"),  # Y
})


@ns.route('')
@ns.param("invreq_id", "资产账户ID")
class GetBondOverviewData(Resource):
    @ns.doc("得到账户债券总览信息数据")
    @ns.response(200, "账户债券信息总览", ns.model("两种债券的数据", {
        "creditOverview": fields.Nested(bond_overview_data, description="信用债"),
        "rateOverview": fields.Nested(bond_overview_data, description="利率债"),
    }))
    @login_require()
    def get(self, user: User, invreq_id: str):...
```



Write APIs as code

Auto generated doc based on code

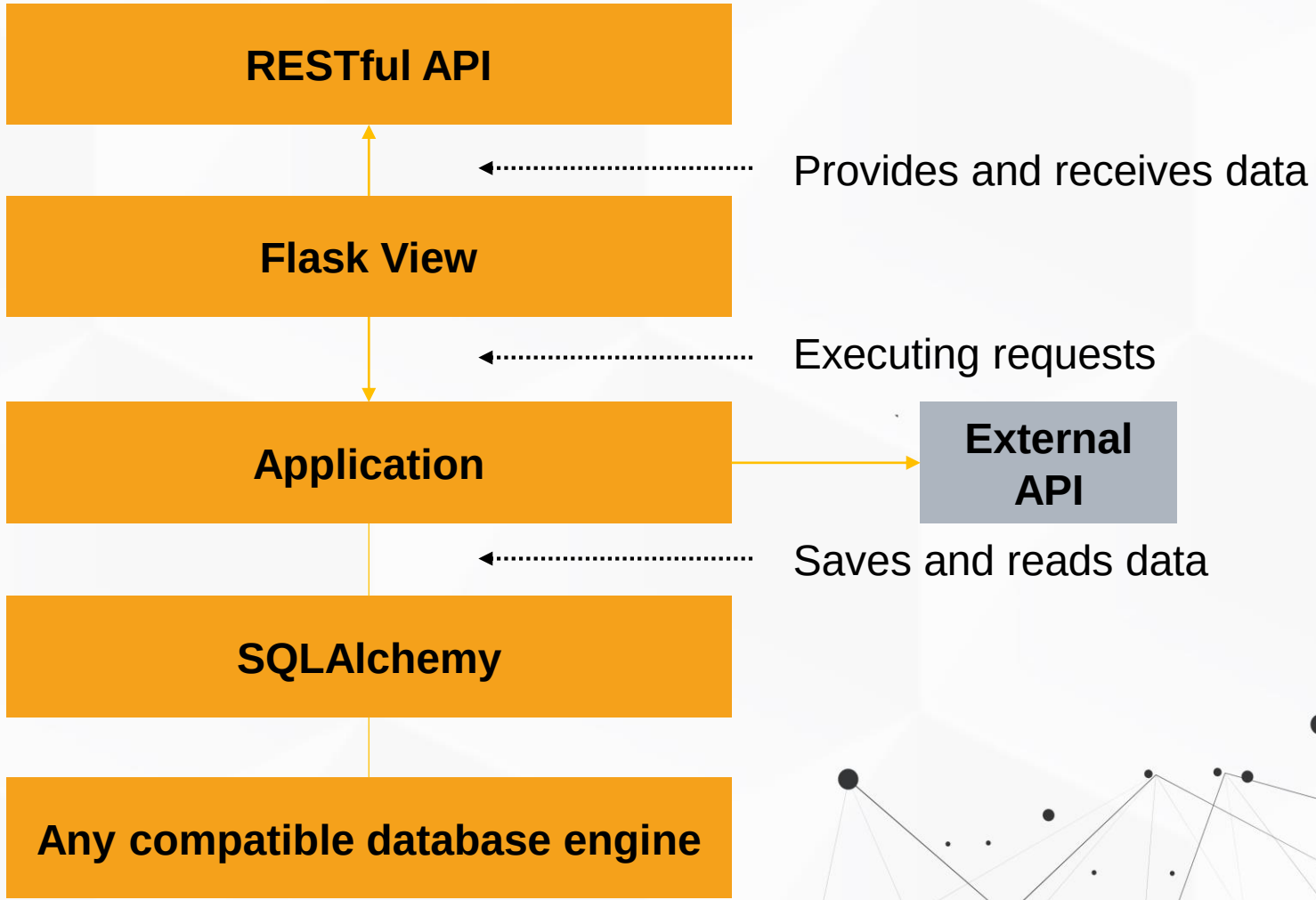
Always up to date

Backend Architecture

Layered Architecture

Each layer is specialized

Help development, debugging and testing



Backend Architecture


```python
from flask import request
from flask_restplus import Resource, Namespace, fields
from decorator.RoleRequest import login_require

from model.user.User import User

ns = Namespace('user/info', description='关于用户信息')

user_profile = ns.model("用户信息", {
    "username": fields.String(description="用户名"),
    "email": fields.String(description="电子邮箱"),
    "registerDate": fields.DateTime(description="注册时间"),
    "avatarUrl": fields.String(description="头像地址")
})

@ns.route('')
@ns.response(200, 'OK')
@ns.response(404, 'user not found')
@ns.response(403, 'access denied')
@ns.response(500, 'system error')
class User(Resource):

    @ns.doc("获得用户信息")
    @ns.response(200, "获得用户信息", user_profile)
    @login_require()
    def get(self, user: User):...


    @ns.doc('修改')
    @ns.expect(user_profile)
    @ns.response(200, "修改成功" )
    def put(self, user: User):...
```

API definition with flask-restplus

Flask Web Framework

Light weight

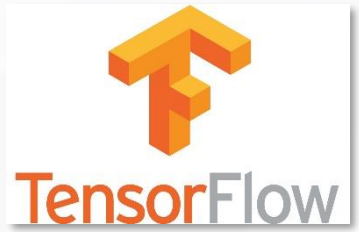
Easy to use

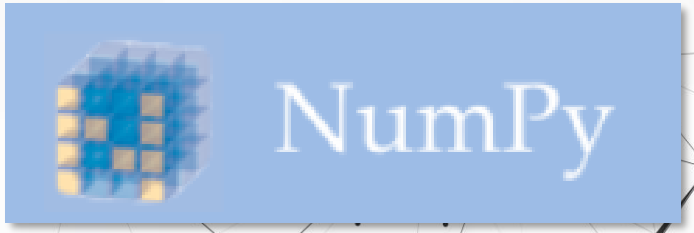
Rich ecosystem

Backend Architecture

```python
from sklearn.cluster import KMeans, DBSCAN
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import LocalOutlierFactor
import pandas as pd
from sklearn.metrics import classification_report
from sklearn.externals import joblib

from factory import DaoFactory


# 训练一个kmeans模型
def train_kmeans():
    # read data
    people = pd.read_csv("data/train_kmeans.csv", header=None)
    kmeans = KMeans(n_clusters=5, random_state=0)
    # outlier detection
    clf = LocalOutlierFactor(n_neighbors=20, contamination=0.01)
    outlier_pre = clf.fit_predict(people)
    normal_people = people[outlier_pre == 1][:]

    # kmeans
    kmeans.fit(normal_people)
    kmeans_pre = kmeans.predict(people)

    # kmeans report
    target_names = ['class 1', 'class 2', 'class 3', 'class 4', 'class 5']
    y_true = [3] * 500 + [0] * 500 + [4] * 500 + [1] * 500 + [2] * 500
    print(classification_report(y_true, kmeans_pre, target_names=target_names))
    joblib.dump(kmeans, "data/kmeans_model.m")
    return kmeans
```

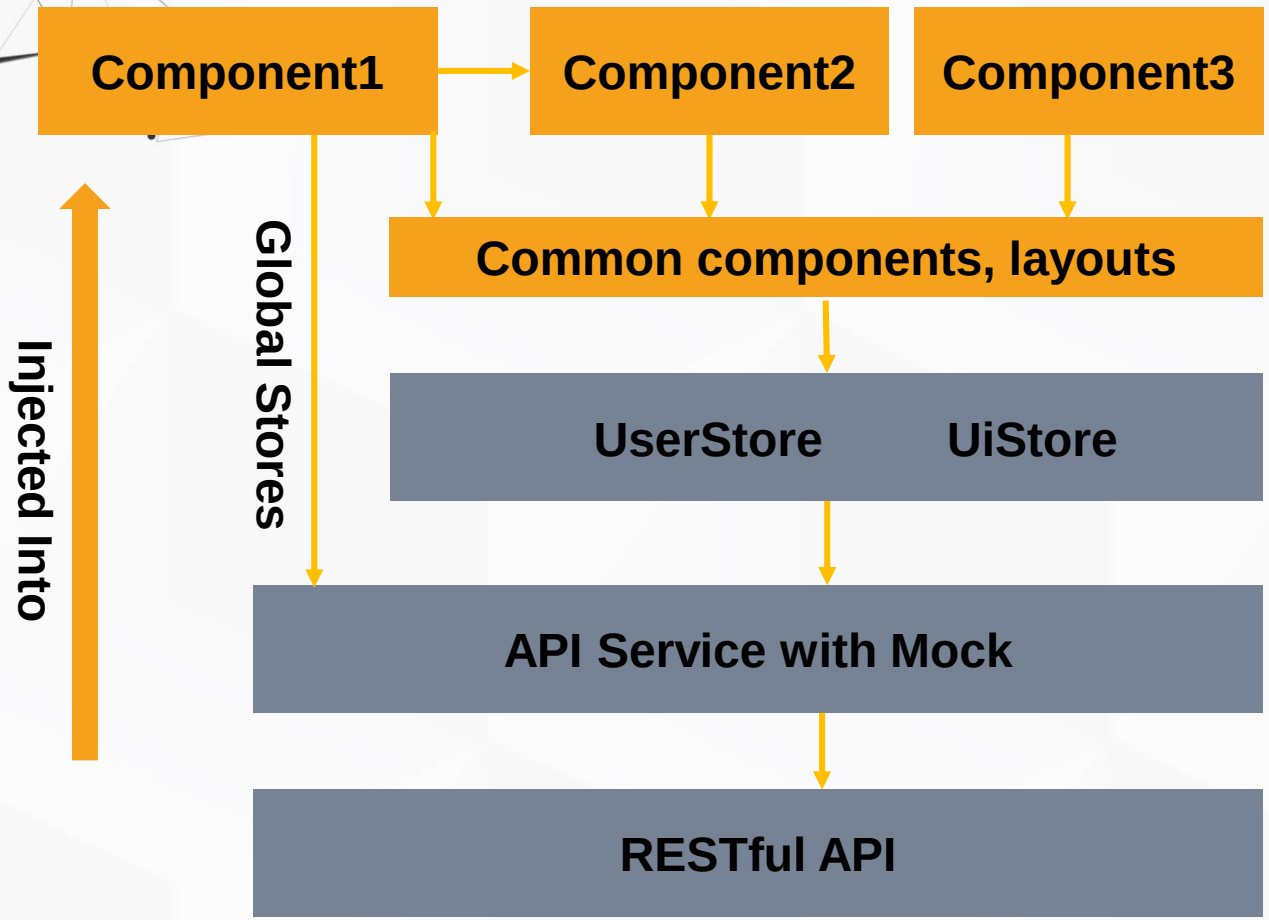
Integration with ML tools





(And More)

Frontend Architecture



Single Page Application

React stack

Most welcomed in the world
Large and evolving community

Code Quality Control

TypeScript with TSLint
Code Splitting
MobX state management
Dependency Injection
API Layer with Mock

Frontend Architecture

UI | Architecture | Workflow

```
interface Props {
  data: Matching[];
}

export class CurrentPosition extends React.Component<Props, {}> {

  render() {

    this.props.data[0].

    return (
      <Card style={{

        <MatchingPag
```

```
percentage Matching (MatchingLis…  number
quotaId Matching (MatchingList.t… string
quotaName Matching (MatchingList… string
totalValue Matching (MatchingLis… number
```

TypeScript

Write elegant code in elegant way
Code style check with TSLint

Frontend Architecture


```
import { KnownRouteConfig, RouteType } from "../routing/RouteConfig";
import routeIndexPage from "../routing/RouteIndexPage";
import { FunctionLayout } from "../layouts/FunctionLayout";

const routes = [
  { type: RouteType.Async, path: "user", component: import("./UserPage") },
  { type: RouteType.Async, path: "help", component: import("./HelpPage")},
  { type: RouteType.Async, path: "invreq", component: import("./InvReqPage") },
  { type: RouteType.Async, path: "quotation", component: import("./QuotationPage")},
] as KnownRouteConfig[];

export default routeIndexPage(routes, FunctionLayout);
```

```
                    55.css   26.5 KiB       55, 6  [emitted]
             55.86cc8.js.   93.3 KiB       55, 6  [emitted]
                    56.css     95 KiB          56  [emitted]
          vendors~main.86cc8.js   1.26 MiB          56  [emitted]
                    57.css   14.8 KiB          57  [emitted]
             57.86cc8.js.   69.5 KiB          57  [emitted]
                    58.css   28.3 KiB          58  [emitted]
             58.86cc8.js.   37.9 KiB          58  [emitted]
            index.86cc8.js.   2.99 KiB          59  [emitted]
                  logo.png   57.8 KiB              [emitted]
                 index.html  828 bytes              [emitted]
```

- Code Splitting
- 50+ small chunks
- Load on demand
- Reduce unnecessary network traffic

Frontend Architecture


```
@Injectable
export class NotificationStore {
  @observable.ref notifications: KnownNotification[] = [];

  @observable refreshing: boolean = false;
  @observable notificationDrawerShown: boolean = false;

  private refreshTimer: NodeJS.Timer;

  @action toggleNotificationDrawerShown = () => {...}

  @computed get count() {...}

  constructor(@Inject private notificationService: NotificationService) { }

  @action refresh = async () => {...}

  @action removeNotification = async (id: string) => {...}
```

MobX
Complex state management made easy

```
@observer
export class Header extends React.Component<Props, {}> {

  @Inject routerStore: RouterStore;
  @Inject navStore: NavStore;

  @action collapse = () => {
    this.navStore.sidebarCollapsed = !this.navStore.sidebarCollapsed;
  }

  to = (path: string) => {
    this.routerStore.push(path);
  }

  toHome = () => {
    this.routerStore.push("/");
  }
```

Dependency Injection
Loosely coupled, flexible and maintainable codebase

Frontend Architecture


```
export default function(useMock: boolean) {
  return [
    {provide: UserService, useClass: useMock ? UserServiceMock : UserService},
    {provide: HttpService, useClass: HttpService},
    {provide: NotificationService, useClass: useMock ? NotificationServiceMock : NotificationService},
    {provide: InvreqService, useClass: useMock ? InvreqServiceMock : InvreqService},
    {provide: QuotationService, useClass: useMock ? QuotationServiceMock : QuotationService},
    {provide: StockAnalysisService, useClass: useMock ? StockAnalysisServiceMock : StockAnalysisService},
    {provide: BondAnalysisService, useClass: useMock ? BondAnalysisServiceMock : BondAnalysisService},
    {provide: ProductAnalysisService, useClass: useMock ? ProductAnalysisServiceMock : ProductAnalysisService},
  ] as Binding[];
}
```

API Service layer:
use mock data during development
by just changing the value of one variable

Teamwork and Modern Workflow

Git

842 commits | 12 branches | 0 releases | 1 environment | 16 contributors | MIT

Hierarchical branch and branch-based workflow

Development Stage	Branch
Development	Team member’s own branch
Review & Test	Frontend or Backend
Integration test	Software
Deployment	Master