

## EVERGREEN Coding Challenge

Damit wir ein Bild von Deinen aktuellen Skills erhalten, möchten wir Dich bitten die nachfolgend beschriebene Coding Challenge zu absolvieren.

### Anforderung:

Erstelle eine Webanwendung die es dem User ermöglicht seine persönliche Risikokennzahl (Wert zwischen 1 und 10) zu ermitteln. Dazu muss der Nutzer 5 Fragen beantworten. Mit den Daten muss ein Endpunkt abgefragt werden. Dieser Endpunkt liefert die ermittelte Risikokennzahl und einige weitere Informationen zur Kennzahl zurück. Diese Informationen sollen dem User als Ergebnis dargestellt werden.

### Optionale Erweiterung:

Der User hat die Möglichkeit sein Ergebnis zu speichern und über eine eindeutige URL immer wieder aufzurufen. Das Speichern der Ergebnisse erfolgt in einer Datenbank und jedes Ergebnis hat einen eindeutigen Identifier.

### Endpunkte:

Zur Ermittlung der Risikoklasse kann dieser Endpunkt verwendet werden:

[https://sandbox.onboarding-api.evergreen.de/api/#/Risk%20Rate/RiskRateController\\_calcRiskRate](https://sandbox.onboarding-api.evergreen.de/api/#/Risk%20Rate/RiskRateController_calcRiskRate)

Die dazugehörigen Fragen sind hier dokumentiert:

<https://documentation.api.evergreen.de/#/onboarding/risk-rate?id=risikofragen>

Hierbei handelt es sich um unsere Partner-API. Du kannst die Swagger Dokumentation verwenden um die API zu testen und zu verstehen. Um Daten aus der Anwendung abzurufen, musst Du einen POST Request an `https://sandbox.onboarding-api.evergreen.de/risk-rate/calculate` senden.

#### Curl

```
curl -X 'POST' \
  'https://sandbox.onboarding-api.evergreen.de/risk-rate/calculate' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "goal": "Retirement",
    "age": "Below56",
    "selfTest": 5,
    "duration": "Below5",
    "behaviour": "KeepCool"
  }'
```

In der Serverantwort erhältst Du die ermittelte Risikokennzahl. Diese muss dem User angezeigt werden. Zusätzlich sollten die Zusammensetzung aus den beiden EVERGREEN Fonds (`riskValues.yin` und `riskValues.yang`) sowie der zu erwartende Ertrag pro Jahr (`riskValues.return`) und die zu erwartende Volatilität (`riskValues.volatility`) angezeigt werden. Bei den Werten handelt es sich um Prozente.



```
Code    Details
201
Undocumented Response body
{
  "answers": {
    "goal": "Retirement",
    "age": "Below56",
    "selfTest": 5,
    "duration": "Below5",
    "behaviour": "KeepCool"
  },
  "calculatedRiskRate": 5,
  "riskValues": {
    "yin": 60,
    "yang": 40,
    "return": 3.2,
    "volatility": 5.9,
    "suretyZone": 45,
    "lowerLimit": 87
  }
}
```

### Design und technische Anforderungen:

Du bist vollständig frei in der Wahl der technischen Umsetzung. Einzige Anforderung ist, dass es mit React für das Frontend umgesetzt wurde.

Bei Design gibt es nur die Vorgabe, dass es auf allen Bildschirmgrößen sauber dargestellt wird. Bezüglich der Farben und Gestaltung kannst Du dich an [evergreen.de](https://www.evergreen.de) orientieren, musst Du aber nicht. Erstelle ein aus Deiner Sicht ansprechendes Design und eine gute und ansprechende Nutzerführung.

### Abgabe:

Bitte stelle uns die Anwendung in einem git-Repository zur Verfügung und nutze dieses ebenfalls zur Dokumentation der notwendigen Schritte zum Starten der Webanwendung.

Bitte setze die Aufgabe nach bestem Wissen und entsprechend Deiner Kenntnisse um. Solltest Du Fragen haben, dann schick diese bitte an [tb@evergreen.de](mailto:tb@evergreen.de).

Wir freuen uns auf Dein Ergebnis!