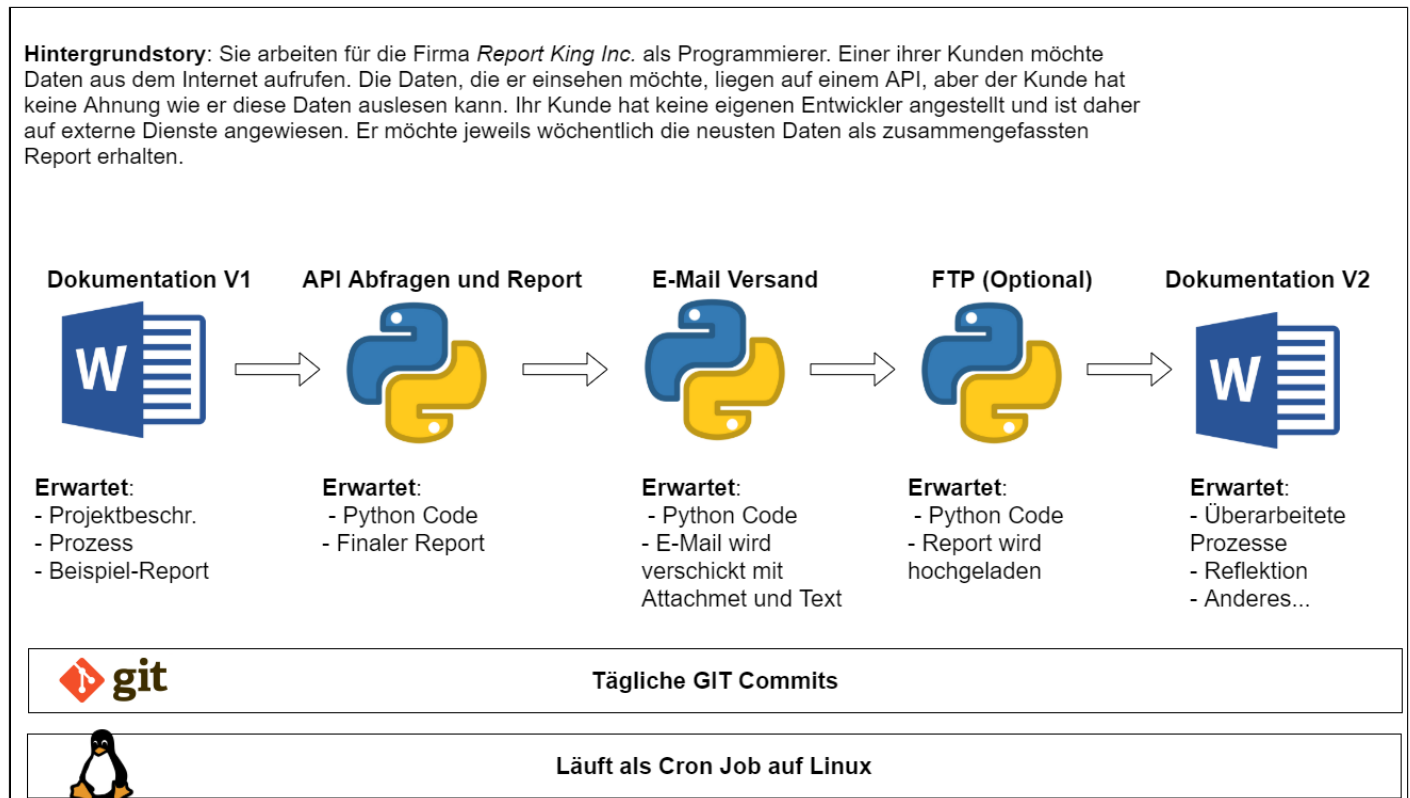


# LB 2 – Projekt

**Hintergrundstory:** Sie arbeiten für die Firma *Report King Inc.* als Programmierer. Einer ihrer Kunden möchte Daten aus dem Internet aufrufen. Die Daten, die er einsehen möchte, liegen auf einem API, aber der Kunde hat keine Ahnung wie er diese Daten auslesen kann. Ihr Kunde hat keine eigenen Entwickler angestellt und ist daher auf externe Dienste angewiesen. Er möchte jeweils wöchentlich die neusten Daten als zusammengefassten Report erhalten.

## Übersicht



## Auftrag

Sie erstellen ein Skript, welches - für Entwickler - typische Aufgaben durchführt. Ein API (Application Programming Interface) aus dem Internet wird angesteuert und Daten daraus in einem Report verarbeitet und dem Kunden zugestellt. Sie werden Anforderungen und Prozesse dabei zuerst in einer Dokumentation definieren und festhalten. **Ihr Skript wird auf einem Linux Rechner als Cronjob ausgeführt.**

## Teil 1 - Dokumentation

Sie beginnen mit ihrer Dokumentation, in der sie die Anforderungen und Abläufe ihres Projekts festhalten. Hier definieren sie ihr Projekt und bestimmen bereits die Basis für ihren Prozess. Dies bedeutet, dass sie sich in das API einlesen müssen, welches sie verwenden werden (Teil 2). Ebenfalls festhalten sollen sie wie ihr Report aussehen soll (ebenfalls Teil 2) mit Beispiel-Daten. Erstellen sie den Beispiel-Report **vor** der Umsetzung.

## **Teil 2 (Hauptteil) – API-Ansteuerung und Verarbeitung in einem Report**

Sie können ein beliebiges API im Internet ansteuern, Daten daraus ziehen und verarbeiten.

### **API Beispiele**

**Idee 1:** Wetter Daten anzeigen (einfach – mittel)

**Idee 2:** Zahlungsverkehr anzeigen aus Stripe.com (mittel-schwer)

**Idee 3:** Zahlungsverkehr anzeigen aus Paypal (mittel-schwer)

**Idee 4:** Crypto Marktdaten anzeigen mit CoinMarketCap (mittel)

**Idee 5:** Spotify Playlist Export (eher schwer)

**Idee 6:** Discord Channel Export (eher schwer)

**Weitere:** Sie sind grundsätzlich frei in ihrer Wahl des API. Viele API kriegen sie gratis oder teilweise gratis, wenn sie nur wenige Requests absetzen. Stellen sie sicher, dass die Anzahl reicht für unsere Zwecke.

<https://rapidapi.com/collection/list-of-free-apis>

### **PDF**

Es gibt verschiedene Wege wie sie ein PDF erstellen:

**Direkt:** Es gibt in Python Bibliotheken mit der sie direkt ein PDF erstellen können. **Nachteil:** ist relativ aufwändig und das Design ist nicht sehr schön oder aufwändig umzusetzen.

**HTML:** Die zweite Möglichkeit ist die Verarbeitung via HTML. Mit CSS können sie ansprechende Seiten erstellen und trotzdem dynamisch HTML-Inhalte verändern oder einfügen. **Nachteil:** ist relativ aufwändig

**Andere:** Evtl. finden sie zusätzliche Varianten. Validieren sie die Variante mit der Lehrperson zuerst.

## **Teil 3 – E-Mail-Versand**

Versenden sie ihre Daten nun via E-Mail an einen Empfänger. Dies wäre in ihrem Fall ihr Kunde. Auch hier gibt es verschiedene Möglichkeiten wie sie vorgehen können:

**SMTP:** Verwenden sie einen normalen SMTP-Server und senden sie eine E-Mail ähnlich wie sie es über ihren lokalen E-Mail-Client tun würden. **Wichtig:** Erstellen sie einen temporären E-Mail-Account bei einem Gratis-Dienstleister. (einfach – mittel)

**SendGrid:** Ein Dienstleister, der E-Mails via API verschickt (mittel)

**SendInBlue:** Ein Dienstleister, der E-Mails via API verschickt (mittel)

## **Teil 4 (Optional) – FTP Upload**

Laden sie ihren Report auf einen FTP hoch. Sie können dazu einen Account bei einem Gratis-Hoster erstellen, z.B. <https://bplaced.net>.

## **Abgabe und Arbeitsweise**

Sie geben am **ersten Tag** einen Link auf ihr GIT Repo an ihre Lehrperson ab (Abgabeort, wird durch die Lehrperson mitgeteilt). Ihr Repository enthält **alle Dateien** (Code, Dokument, etc).

Sie werden **mindestens einmal täglich** ihren Fortschritt commiten. Dies wird **überprüft und ist Teil der Benotung**.

# Bewertungs-Kriterien

Es gibt keine Checkliste mit detaillierten Kriterien, wegen der Komplexität des Projektes. Das Gesamtbild und Detaillierungsgrad wird bewertet. Trotzdem finden sie folgend viele Hinweise.

**Allgemeines Vorgehen:** Versuchen sie zuerst eine einfache Variante ihres Scripts zu erstellen, bei dem sie wissen, dass es evtl. noch Fehler hat oder Code Guidelines nicht eingehalten wurde. Wenn ihr Basis-Skript läuft, können sie sukzessive ihren Code verbessern. Z.B. können sie eine einfache PDF-Report-Variante erstellen und anschliessend umstellen auf eine bessere. Auch beim E-Mail-Versand können sie zuerst mit einem normalen SMTP (z.B. Google) arbeiten und wenn ihnen Zeit bleibt, stellen sie um auf einen API-Service.

## Dokumentation

Die Dokumentation soll sicherlich die wichtigsten Bestandteile enthalten (<https://de.wikipedia.org/wiki/Projektauftrag#Bestandteile>) und die Inhalte ästhetisch darstellen. Ebenfalls notwendig sind Anforderungs-/Projekt-Beschreibung, eine graphische (standardisierte, z. B. UML) Darstellung des Ablaufs und eine abschliessende Reflektion ihrer Arbeit.

Je nach Detaillierungsgrad kann ihre Dokumentation besser bewertet werden. Z.B. können sie ein Aktivitätsdiagramm simpel oder komplex gestalten unter der Verwendung der (korrekten) Symbolik (siehe Modul m319).

Sie können auch eine einfache Variante ihres Diagramms vor der Entwicklung erstellen und anschliessend die korrekte, detaillierte nach der Programmierung.

## Code Allgemein

Im Code wird angeschaut, ob sie gewisse Standards einhalten wie inline-Dokumentation, Header, etc. Auch hier soll auf das Modul m319 und deren Inhalte verwiesen werden.

**Achtung:** Sie erstellen hier ein Skript, welches automatisch wöchentlich ausgeführt werden soll. Es darf also keine Kommunikation mit einem Benutzer geben und keine grafischen Oberflächen.

Wichtig ist dafür, dass die Eigenschaften oder externen Variablen wie Benutzernamen, API-Key, etc sich **nicht** im Source-Code befinden, sondern mit Übergabeparameter und/oder Config-Dateien übergeben werden.

Config Files in Python: <https://towardsdatascience.com/from-novice-to-expert-how-to-write-a-configuration-file-in-python-273e171a8eb3>

Beispiel:

```
1 import json
2 import yaml
3
4 def read_json(file_path):
5     with open(file_path, "r") as f:
6         return json.load(f)
```

```
1 {
2     "APP": {
3         "ENVIRONMENT": "test",
4         "DEBUG": true
5     },
6     "DATABASE": {
7         "USERNAME": "xiaoxu",
8         "PASSWORD": "xiaoxu",
9         "HOST": "127.0.0.1",
10        "PORT": 5432,
11        "DB": "xiaoxu_database"
12    }
13 }
```

## ***API und Report***

In diesem Teil wird bewertet, wie komplex ihre API-Abfragen und deren Verarbeitung in den Report sind. Wenn sie z.B. nur eine URL aufrufen und die Inhalte durchlaufen und aufzählen, kriegen sie weniger Punkte, als wenn sie mehrere API-Aufrufe benötigen und die Verarbeitung komplexer ist, z.B. mit mehreren Loops und/oder Selektionen. Finden sie ihren eigenen Komplexitätsgrad, so dass sie auch den folgenden Teil (E-Mail-Versand) noch schaffen. **Auch hier als Richtlinie:** Arbeiten sie vielleicht zuerst einfach und erweitern sie ihren Report anschliessend, falls sie Zeit haben.

Ebenfalls bewertet wird die Form wie ihr Report erstellt wird und anschliessend aussieht. Die Optionen wurden oben bereits erklärt.

API Requests mit dem Modul «Requests»: [https://www.w3schools.com/python/module\\_requests.asp](https://www.w3schools.com/python/module_requests.asp)

API Requests mit dem Modul «Urllib3»: <https://pypi.org/project/urllib3/>

PDF erstellen mit dem Modul «fpdf»: <https://towardsdatascience.com/creating-pdf-files-with-python-ad3ccadfae0f>

PDF erstellen via HTML mit dem Modul «xhtml2pdf»: <https://xhtml2pdf.readthedocs.io/en/latest/usage.html>

## ***E-Mail-Versand***

In diesem Teil wird bewertet, welche Variante sie zum Versenden der E-Mails verwenden.

Ebenfalls relevant ist, dass der Anhang (Report) mitgeschickt wird und der E-Mail-Text korrekt ist. Erinnern sie sich an das Szenario: Sie schicken einem Kunden einen Report zu, der Text soll entsprechend fehlerfrei und eindeutig betreffend Inhalt sein. Verwenden sie nicht ihren privaten oder TBZ E-Mail Account. Es gibt viele Gratisanbieter wie Gmail.

Send E-Mail with Python via Gmail: <https://geekflare.com/send-gmail-in-python/>

Send E-Mail with Python via sendgrid: <https://docs.sendgrid.com/for-developers/sending-email/v3-python-code-example>

Send E-Mail with Python via sendInBlue: <https://developers.sendinblue.com/reference/sendtransacemail>

## ***FTP Upload (Optional)***

In diesem Teil können sie relativ einfach zusätzliche Punkte holen, falls ihnen etwas anderes nicht gelingen möchte. Bewertet hier wird, ob ihr Report korrekt hochgeladen wird. Auch hier gilt wieder, dass sie - gemäss Szenario – den Report auf den FTP-Server eines Kunden hochladen. Die Datei soll entsprechend benannt und archiviert werden.

Upload Files via FTP: <https://www.geeksforgeeks.org/how-to-download-and-upload-files-in-ftp-server-using-python/>