2017

Python Webframeworks

Django

*Yunus Sari – Muhammedhizir Kanyildiz*

Inhaltsverzeichnis

[1 Was ist Django? 1](#_Toc477099454)

[1.1 Erklärung 1](#_Toc477099455)

[1.2 Vorteile 1](#_Toc477099456)

[1.3 Nachteile 2](#_Toc477099457)

[1.4 Beispiele 3](#_Toc477099458)

[2 Installation 5](#_Toc477099459)

[3 Konfiguration 8](#_Toc477099460)

[3.1 Erstellung eines Projektes 8](#_Toc477099461)

[3.2 Ausführen des Django-Servers 8](#_Toc477099462)

[3.3 Klonen eines Projektes 8](#_Toc477099463)

[4 Anwendungsfall 9](#_Toc477099464)

[5 Datensätze 9](#_Toc477099465)

[6 Anlegen/Bearbeiten/Löschen von Datensätzen 9](#_Toc477099466)

[7 Schwierigkeiten 9](#_Toc477099467)

[7.1 Environment Variable 9](#_Toc477099468)

[8 Aufwandschätzung 10](#_Toc477099469)

# Was ist Django?

## Erklärung

Django ist ein in Python geschriebenes Full Stack Framework, dass die schnelle Entwicklung von Web-Applikationen ermöglicht. Dabei wird Wert auf sauberen Code und die Wiederverwendbarkeit der einzelnen Komponenten gelegt. Der Quellcode und die umfangreiche Dokumentation sind unter einer BSD-Lizenz lizensiert. Die Django Software Foundation stellt die Weiterentwicklung von Django sicher.

Django unterstützt durch seine Architektur und Werkzeuge eine schnelle Entwicklung (Rapid Development) von Websites und neuen Komponenten.

Loose Coupling, also die lose Koppelung der verschiedenen Teile des Frameworks und der Applikation, stehen bei Django stark im Vordergrund. Dadurch soll die Qualität und Wiederverwendbarkeit des Codes erhöht werden. [1]

## Model-Template-View

Django ist nach dem Model-Template-View (MTV) Muster aufgebaut. MTV orientiert sich am bekannten Model-View-Controller Muster (MVC).

Der in Django enthaltene Object Relational Mapper (ORM) überträgt die Models in Datenbankstrukturen und führt alle Operationen in der Datenbank durch. Es können alle gängigen Datenbanken benutzt werden. Alle Models werden in Python geschrieben.

Die Template-Engine unterstützt die Vererbung von Templates und bietet umfangreiche Filter und Templates. Diese können auch selbst erweitert werden. Der View holt die Daten, zum Beispiel mit Hilfe des Object Relational Mappers. Es können aber auch anderen Datenquellen genutzt werden. Die URLConf steuert das Routing. Mit Hilfe von regulären Ausdrücken wird der Request dem richtigen View zugewiesen. [2]

## Vorteile

Alle großen Python-Web-Frameworks bieten sich dann an, wenn eine Anwendung in relativ kurzer Zeit, möglichst sauber implementiert werden soll. Auf dieses Gebiet ist bereits Python ausgerichtet. Unterstütz wird dies durch die bereits standardmäßig in Python- Interpretern enthaltenen Module. Viele von ihnen erfüllen bereits wichtige Aufgaben die im Internet benötigt werden und sind mit wenig Einarbeitungszeit leicht zu verwenden.

Auch ist Python nicht properitär, wird also nicht von einem Unternehmen entwickelt, welches seine Interessen bestmöglich durchsetzen will. Sinnvolle Erweiterungen der Sprache werden ständig durchgeführt und erst nach ausgibiger Diskussion übernommen. Hinzu kommt die Plattformunabhängigkeit der Sprache, so dass die Web-Frameworks auf nahezu jedem größeren Betriebssystem laufen.

Für Entwickler ist natürlich eine besonders gute Dokumentation sehr wichtig. Zu allen drei vorgestellten Web-Frameworks ist eine solche, auch in sehr detallierter Form, vorhanden. Die Dokumentationen beziehen sich primär auf das Entwickeln neuer Anwendungen und gehen weniger auf konkrete Details des Systems ein. Dies erleichter vorallem das Erstellen von Applikationen.

Ein besonders wichtiger Punkt ist die große Auswahl an verschiedenen Web-Frameworks. Diese ermöglicht es, das für sich passende zu wählen und zu verwenden. Außerdem bestehen für die größeren Frameworks bereits eine Reihe von vorgefertigten Anwendungen, wie Shops und Gästebücher, die zur Verwendung nur noch eingebunden werden müssen. [3]

## Nachteile

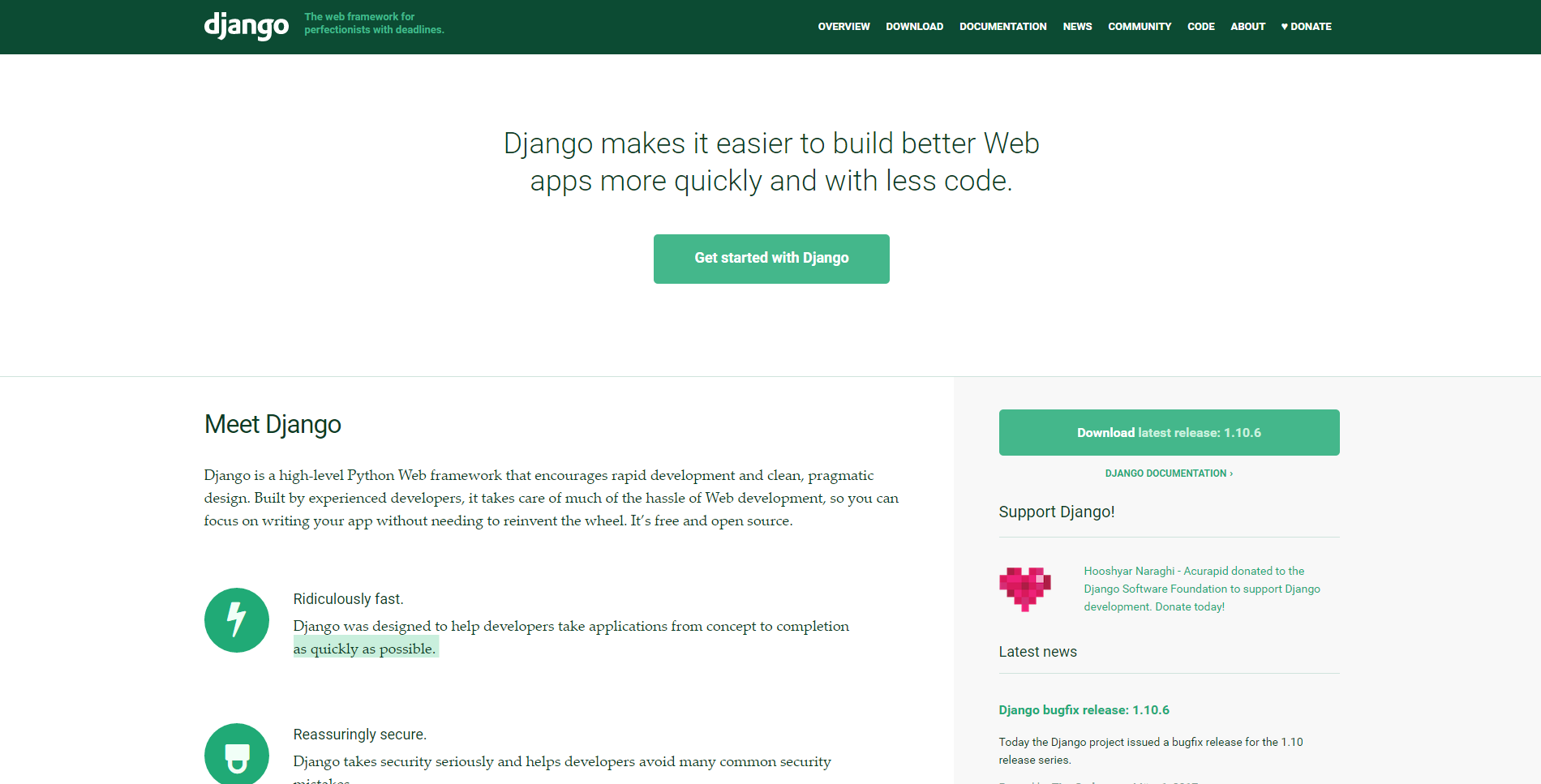
Aber auch das Einbringen von Python, und Anwendungen die in dieser Sprache geschrieben sind, in ein Unternehmen wurde bis vor kurzem erschwert, da nahezu kein Marketing auf Seiten der Python Software Foundation und auch nicht bei den Web-Frameworks bestand. Dies ist aber oftmals ein entscheidender Grund für einen Manager ein bestimmtes Produkt zu verwenden oder nicht, da für diesen Sprach-Konzepte von Programmiersprachen wenig interessant sind.

Auch ist Python ursprünglich nicht primär für das Web ausgelegt, bietet aber schon länger sehr gute Module für diese Aufgabe. Daher konzentriert man sich lediglich auf den logischen Teil von Web-Anwendungen und bietet die Gestaltung der Seiten über die bewährte Technik der Templates an. [3]

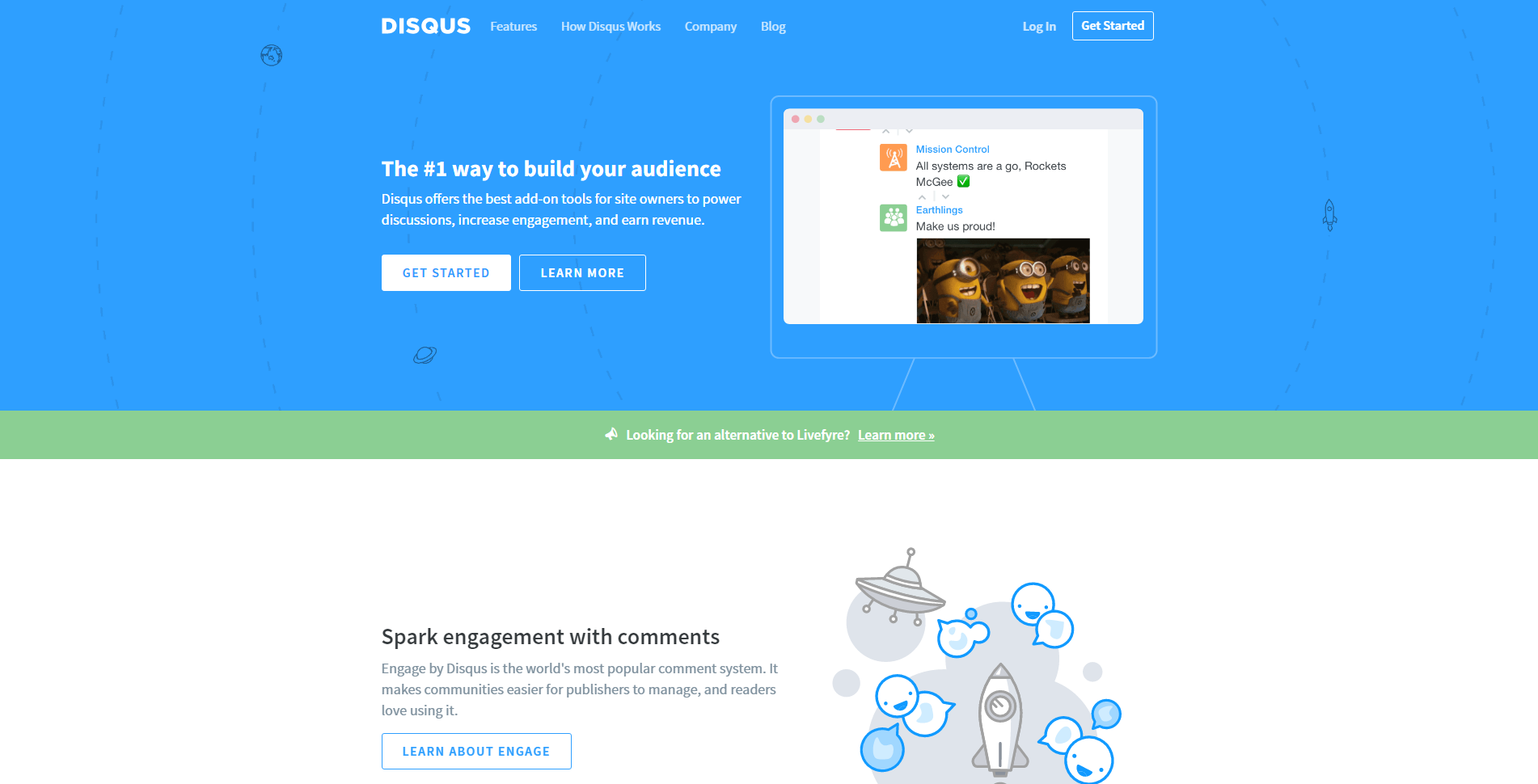
## Beispiele

Beispiele von Webseiten, die Django benutzen:

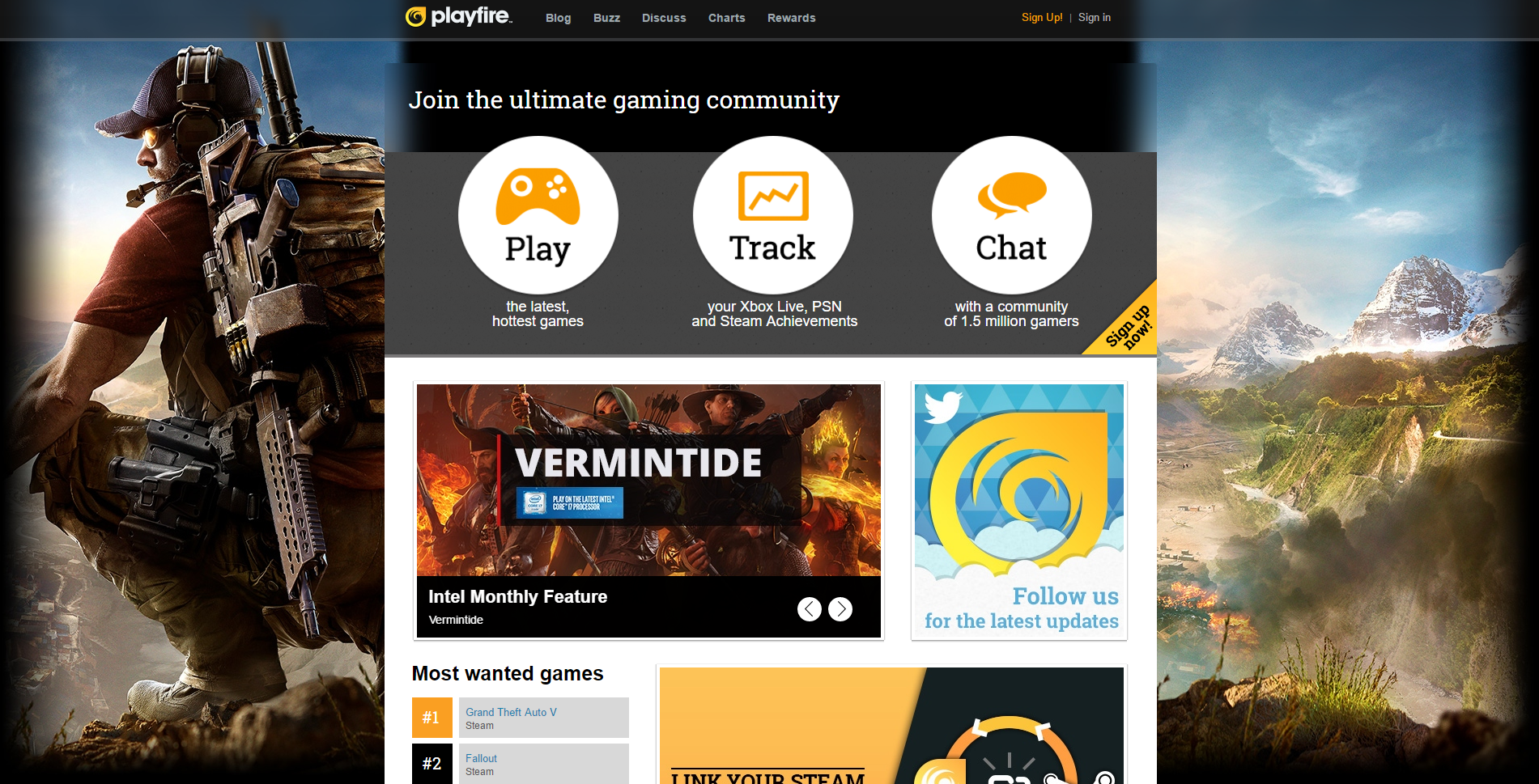
* <https://www.djangoproject.com/start/overview/>



* <https://disqus.com/>



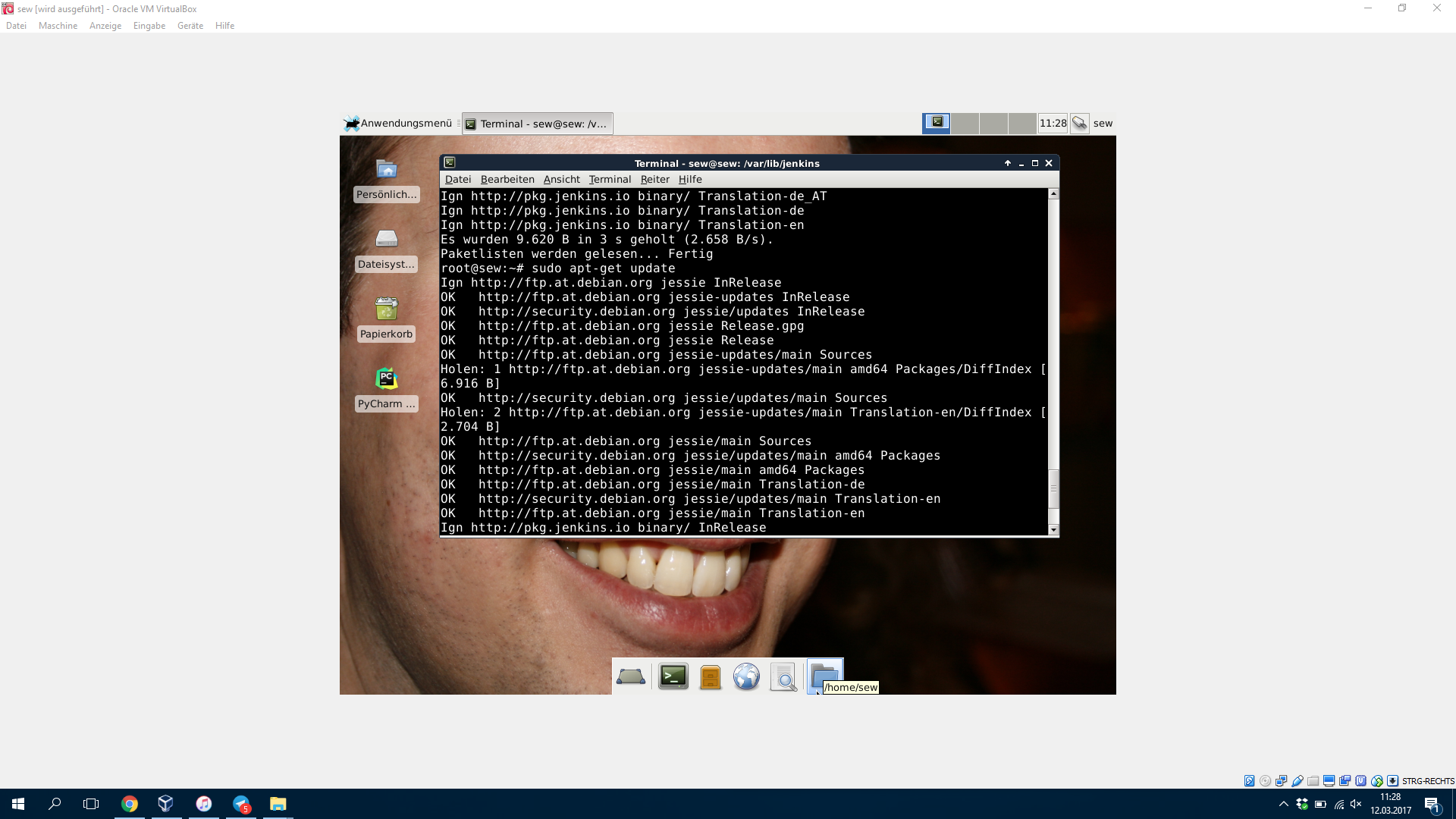
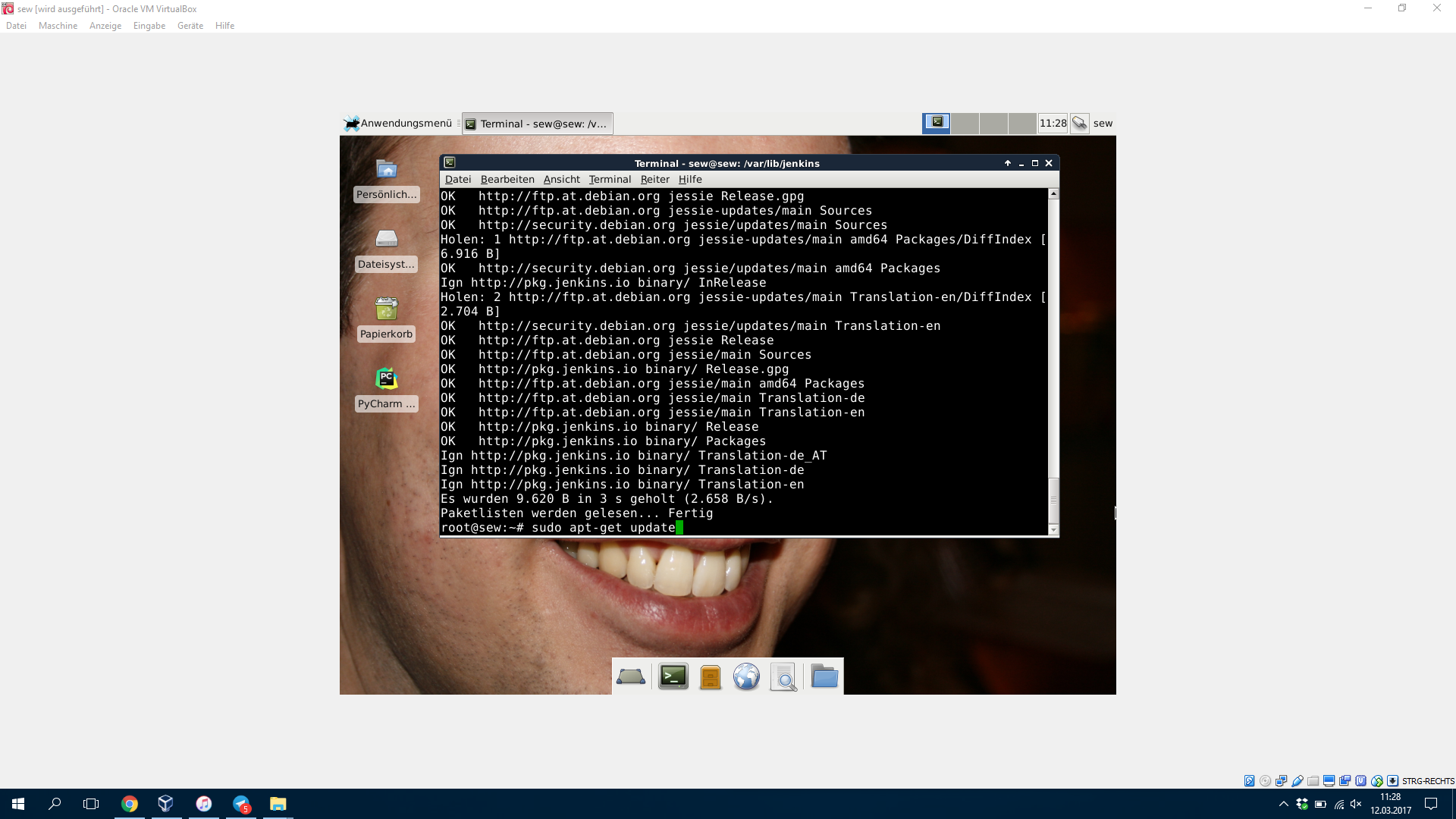
* <https://www.playfire.com/>



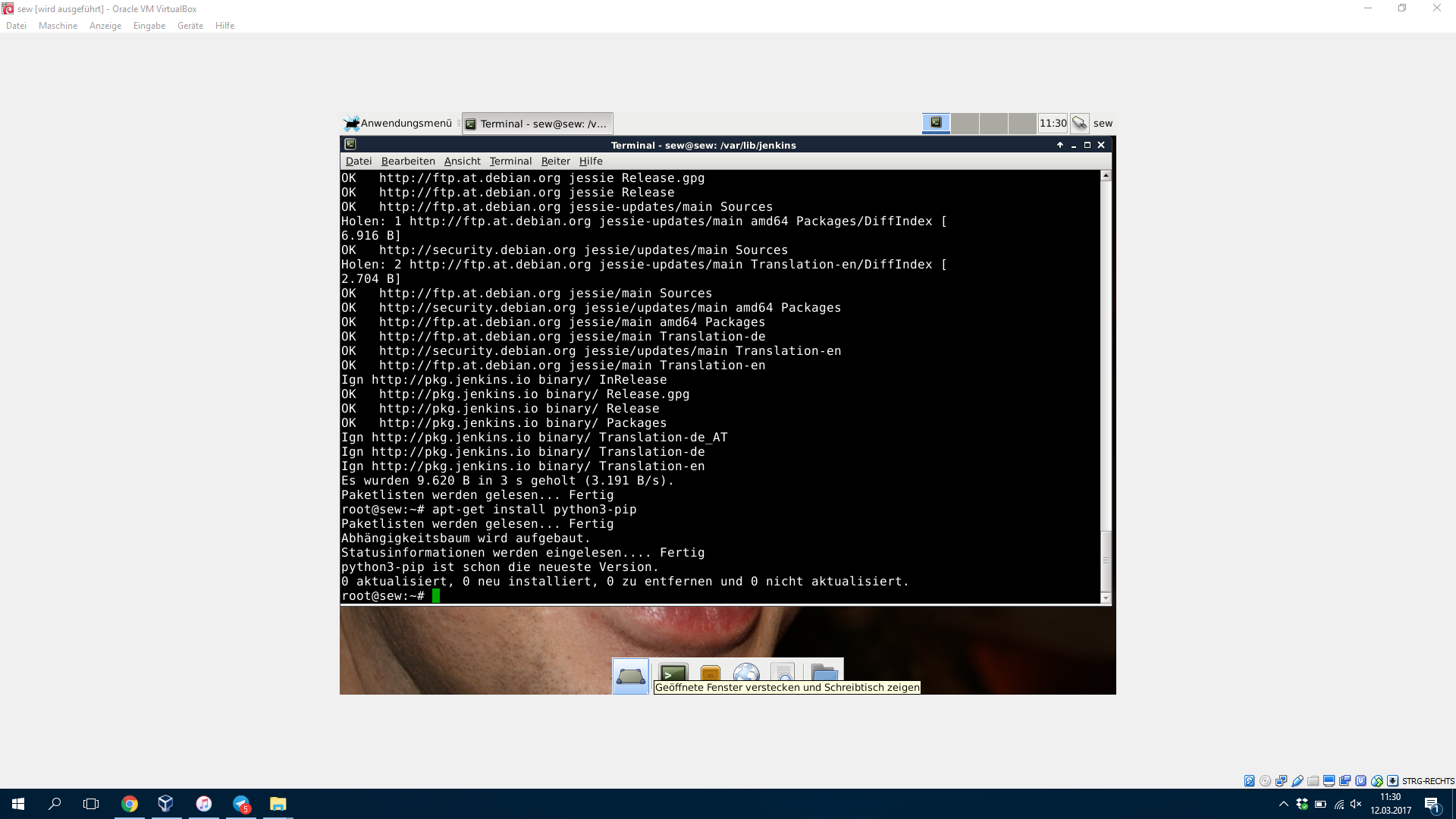
* <https://www.washingtonpost.com/>

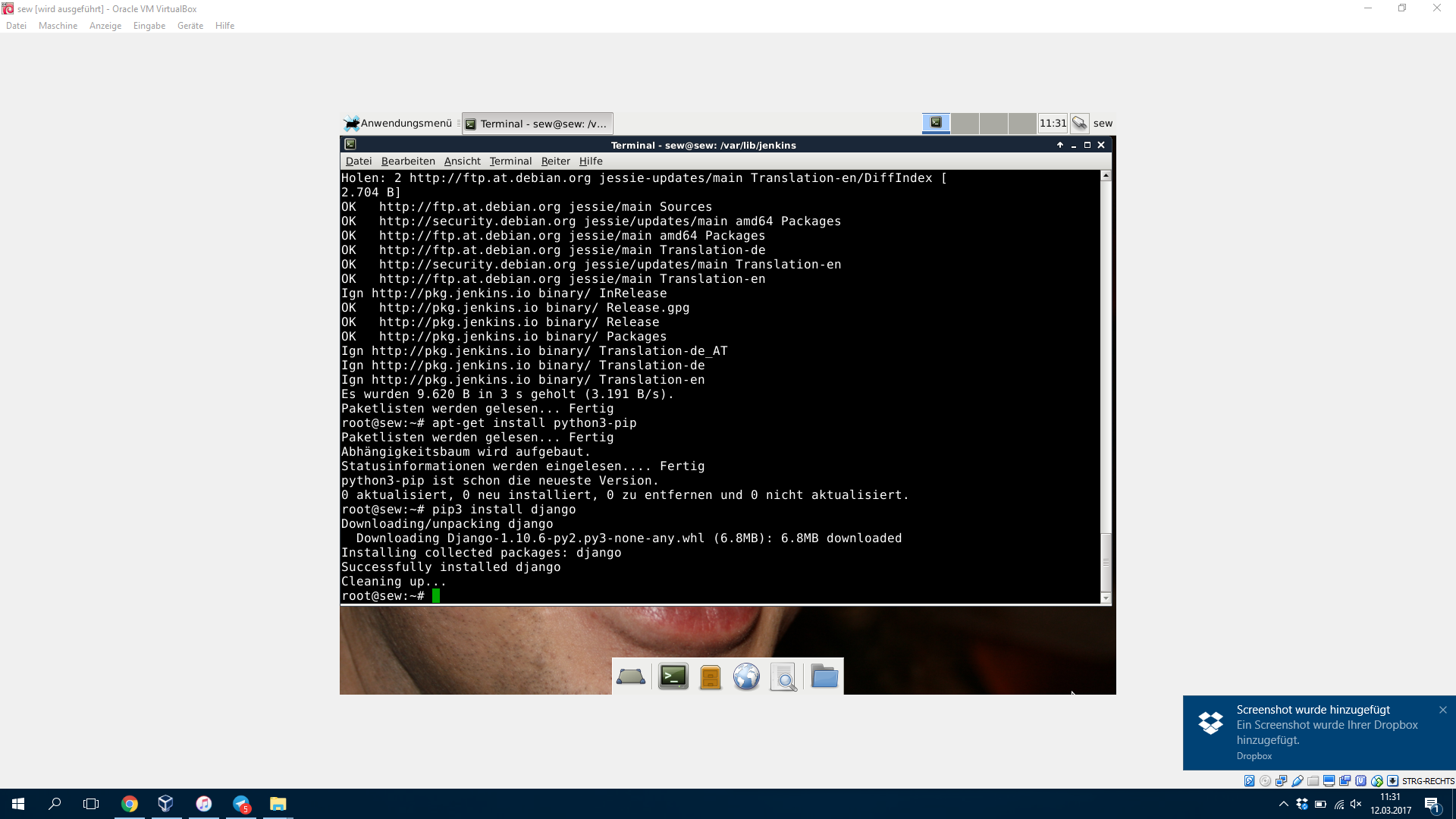
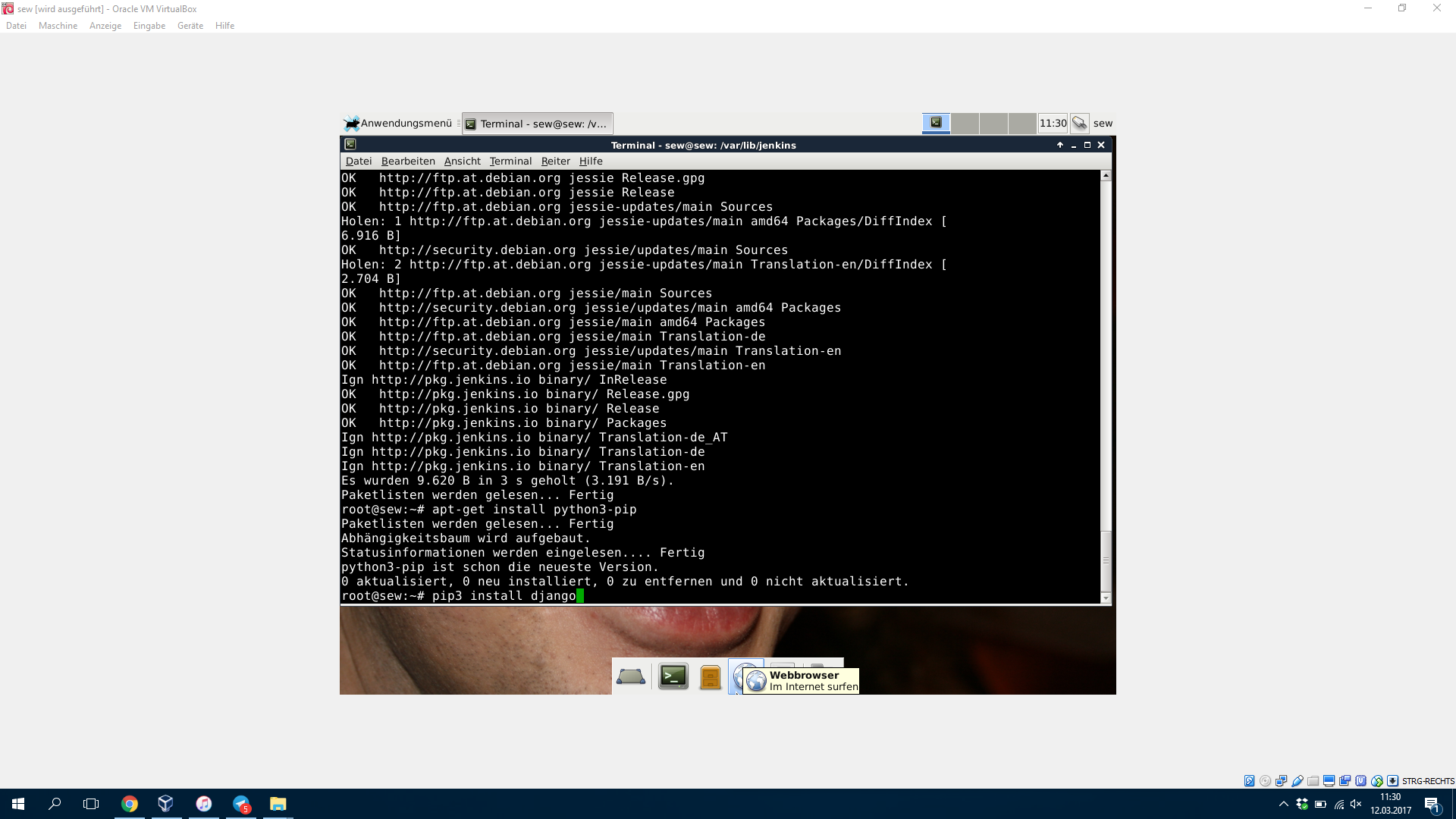


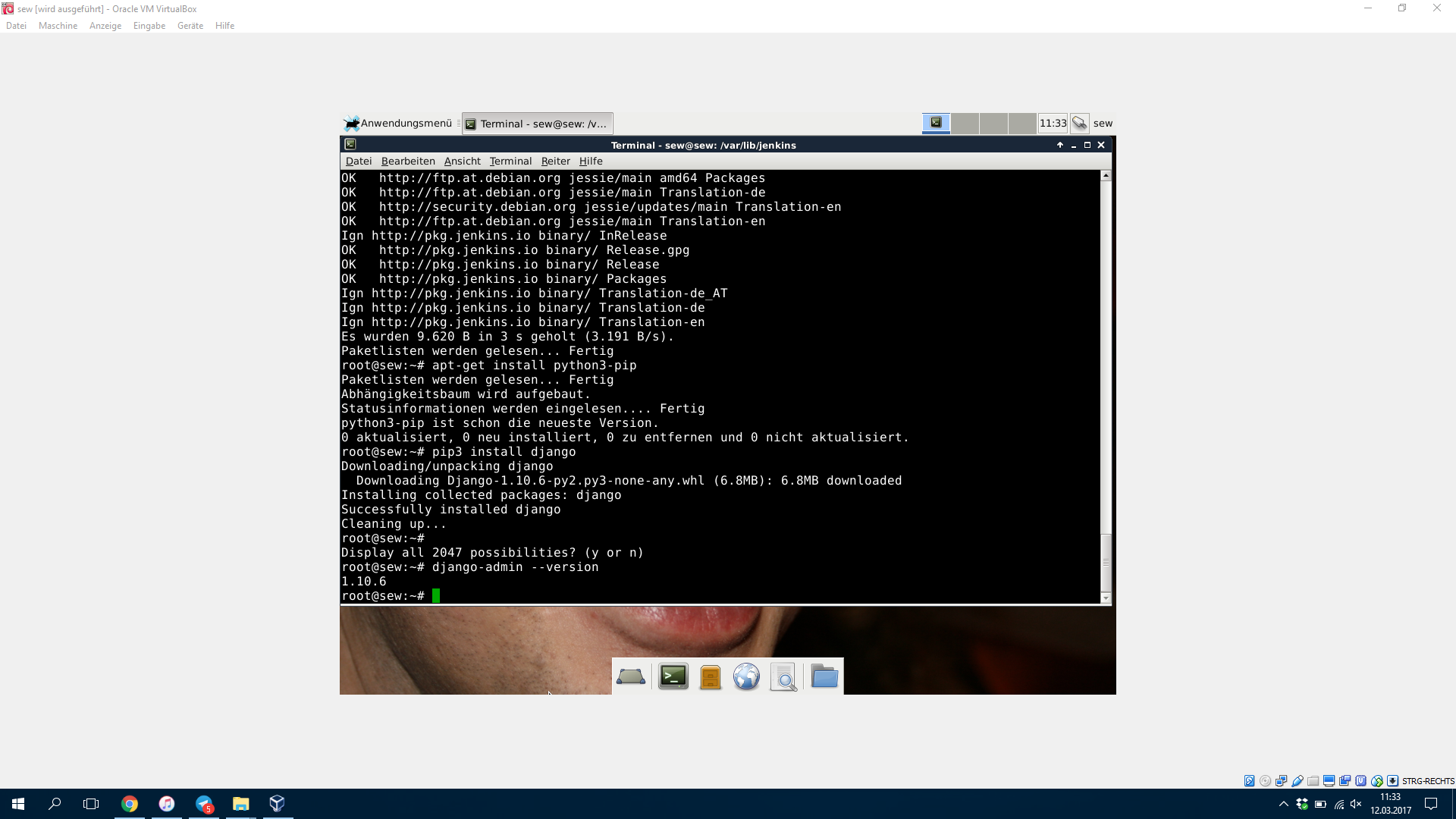
# Installation

Vor der Installation wurde ein Update durchgeführt.

Nach diesem Schritt wurde Pip installiert

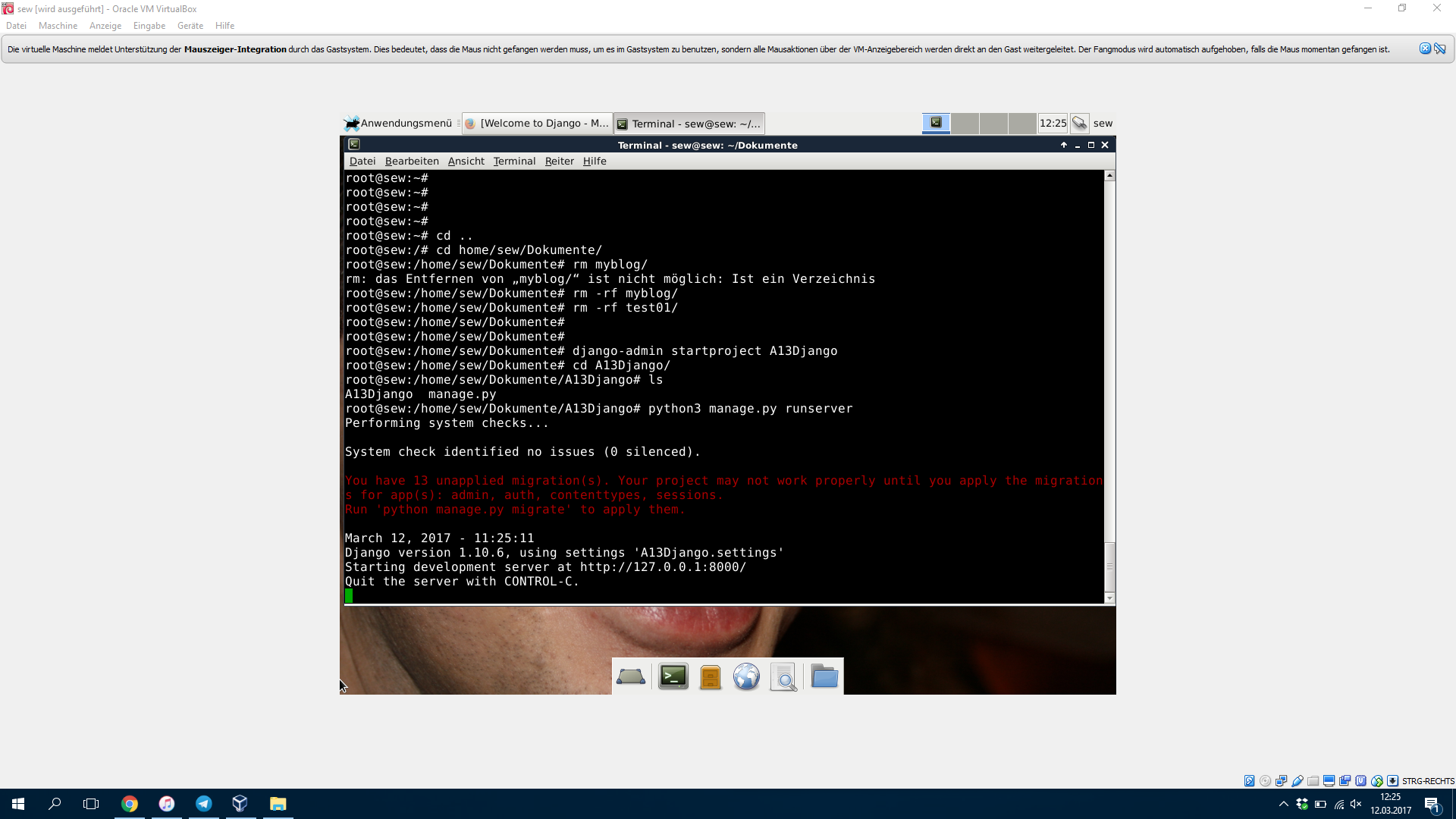


Django wurde installiert.

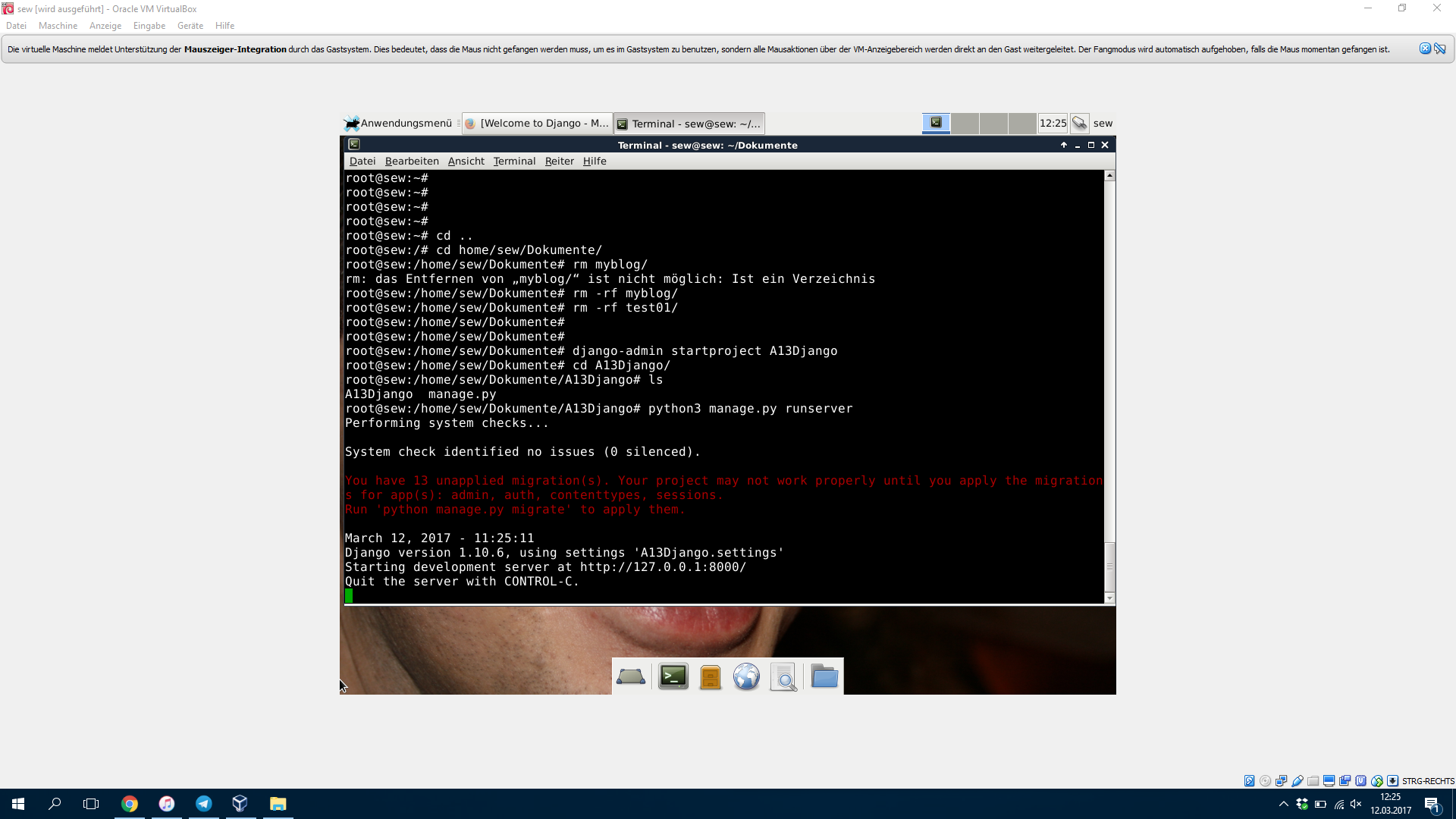
Es wurde überprüft, ob Django installiert wurde

# Konfiguration

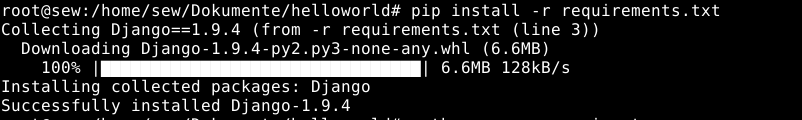
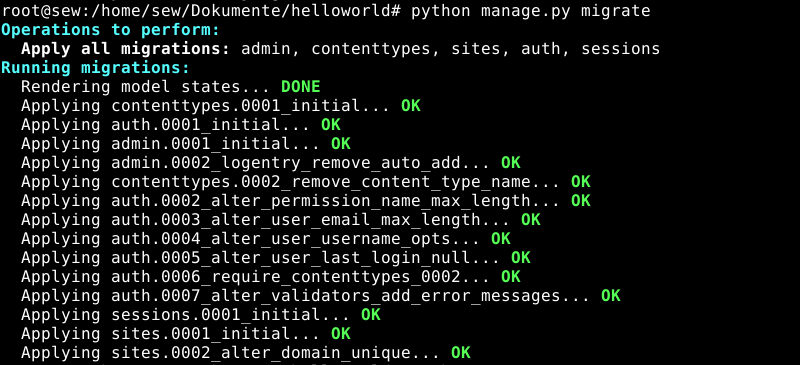
## Erstellung eines Projektes

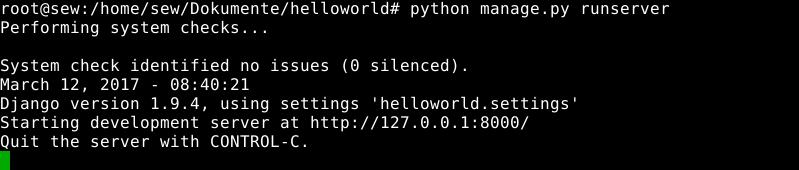


## Ausführen des Django-Servers



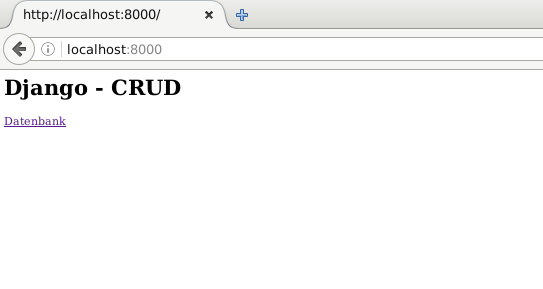
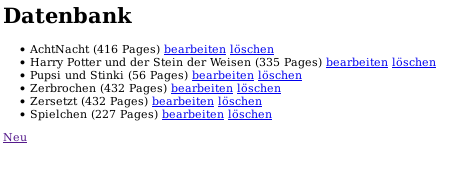
## Klonen eines Projektes

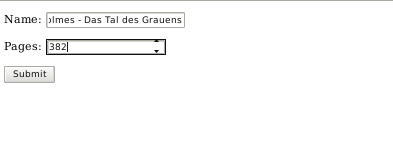
Nach dem Klonen des Projektes wurde folgender Befehl ausgeführt. 

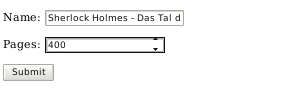
Danach kann der Django-Server ausgeführt werden. 

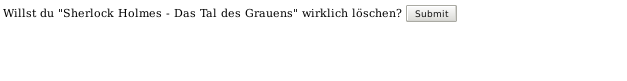
# Beispiel

Startseite

Datenbank

Neuer Eintrag

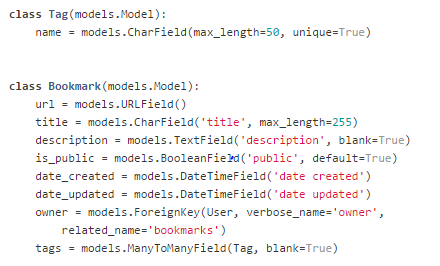
Bearbeiten

Löschen

# Models erstellen

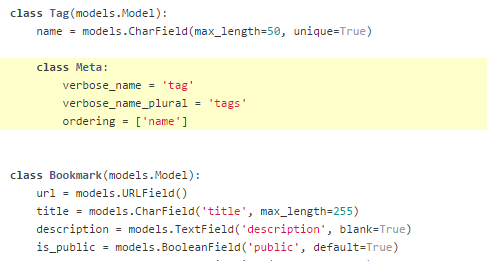
Models definieren, wie Daten gespeichert werden. Üblicherweise repräsentiert ein Model eine Tabelle in der Datenbank und besteht aus Feldern, Metadaten und Methoden. Durch diese Informationen kann Django automatisch eine Schnittstelle zur Datenbank generieren, die es erlaubt, objektorientiert auf diese zuzugreifen. Man spricht hierbei auch von Object-Relational-Mapping (ORM).

# Felder

Felder werden als Klassenattribute definiert und entsprechen den Spalten einer Tabelle. Django bietet verschiedene Typen von Feldern, um die Daten in einem möglichst sinnvollen Format abzulegen. So entspricht beispielsweise CharField einer Spalte vom Typ VARCHAR in einer SQL-Datenbank. Eine [Liste der Feldtypen](https://docs.djangoproject.com/en/1.8/ref/models/fields/) kannst du in der Django-Dokumentation finden.

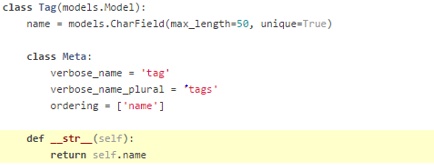
# Metadaten

Models können [Metadaten](https://docs.djangoproject.com/en/1.8/ref/models/options/) enthalten, welche die Darstellung oder das Verhalten beeinflussen. Sie werden in der internen Klasse Meta definiert. Im Beispiel werden der Name für die Darstellung (singular und plural) sowie die Standardsortierung festgelegt.



# Methoden

You can now add Model functionality [Methods](https://docs.djangoproject.com/en/1.8/topics/db/models/#model-methods) for actions that apply to a single record. For instance, it is usual to create a human readable form of the record with the method \_\_str\_\_.



# Aufwandschätzung

|  |  |  |
| --- | --- | --- |
|  | Geschätzter Aufwand | Tatsächlicher Aufwand |
| Installation & Konfiguration | 2 h | 2,5 h |
| Anwendungsfall | 2 h | 2 h |
| Datensätze | 1 h | 0,5 h |
| Anlegen/Bearbeiten/Löschen von Datensätzen | 4 h | 2 h |
| Protokollieren | 1 h | 1 h |
| Summe | 10 h | 8 h |

# Quellenverzeichnis

[1] drweb.de  
<https://www.drweb.de/magazin/framework-django/>

[2] django-workshop.de  
<http://www.django-workshop.de/einfuehrung.html>

[3] t3n.de  
<http://t3n.de/news/django-python-framework-schnelle-entwicklungsprozesse-335967/>