

# Tekniskt Dokument - ExternalsController

FlowerPower (Grupp 10)

Hampus Pukitis Furhoff - *hfurhoff@kth.se*

Datum	Version	Författare	Beskrivning
2018-05-22	1.0	Hampus Pukitis Furhoff	Första version

## Innehåll

1. Bakgrund
2. Syfte
3. Nuvarande design
4. Krav på komponenten
5. Testning
6. Gränssnitt
7. Beroende
8. Intern struktur
9. TODO

## 1. Bakgrund

Vi i grupp 10 har jobbat med en produkt i några veckor nu som vi kallar FlowerPower. FlowerPower är en produkt som ska hjälpa privatpersoner eller företag att ta hand om sina växter genom att avläsa fuktvärden från växter och skicka emailnotis när växten håller på att torka ut. På en hemsida ska man även kunna se dessa fuktvärden samt sätta av och på en bevattningspump. Vi har därför tänkt att på vår server där vi har vår hemsida ska vi även ha ett antal trådar som håller koll på dom olika planterna och det är där som behovet för ExternalsController kommer in. ExternalsController är tänkt att vara den komponent som håller koll på dom olika plant-trådarna och som övriga komponenter måste gå igenom för att kunna utföra operationer på plant-trådarna.

## 2. Syfte

Syftet med detta dokument är att det ska finnas samlad information om komponenten ExternalsController så att vem som helst ska kunna använda komponenten eller utveckla vidare på den.

## 3. Krav på komponenten

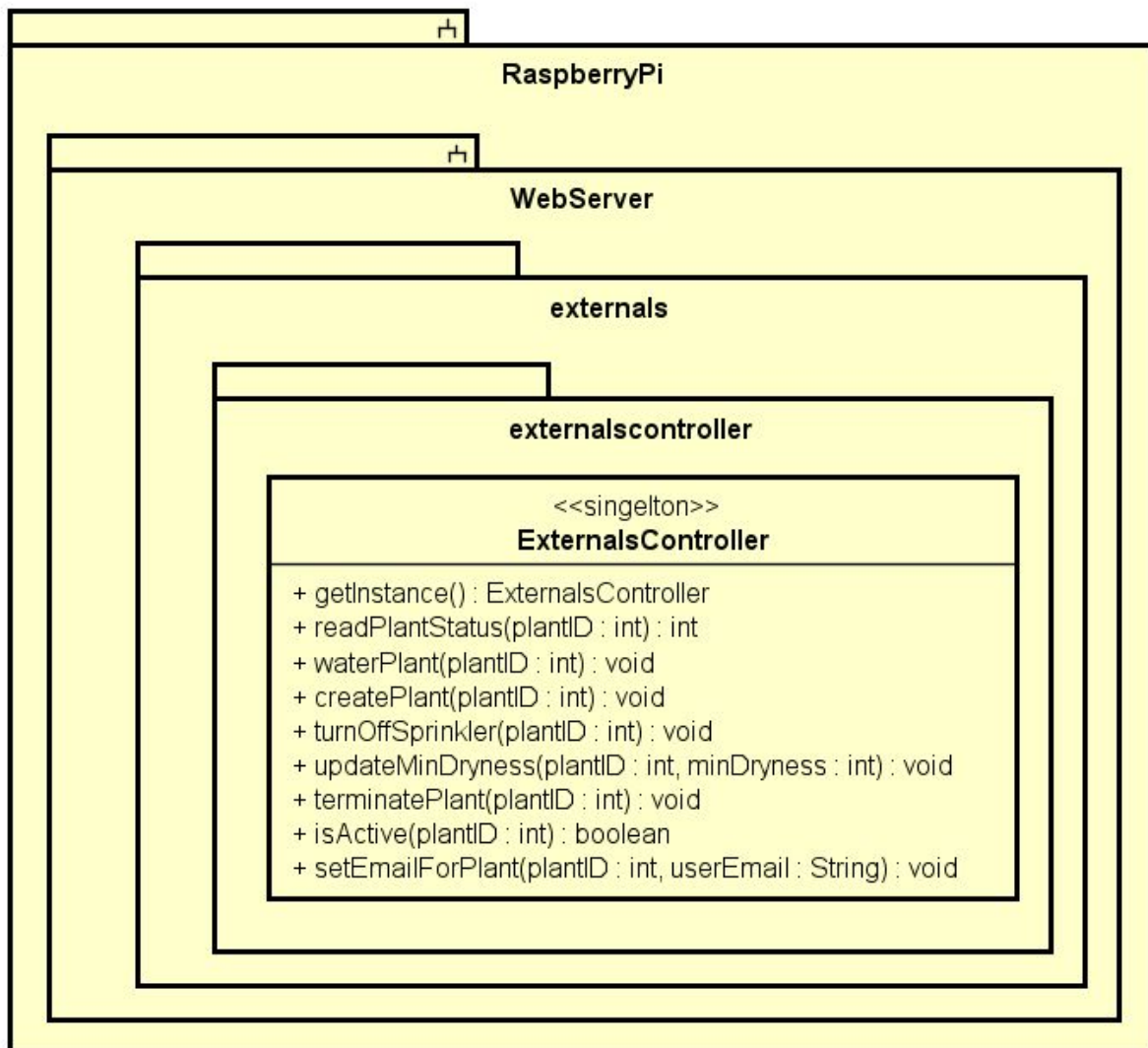
### Användarkrav

Komponenten ska kunna hantera flera plant-trådar och göra operationer på dessa så att någon anropare av dess metoder ska kunna utläsa plantstatus, starta och stoppa plantprocesser, ändra när och till vem notis skickas när plantan är torr, samt styra bevattning.

### Utvecklingskrav

Komponenten skrivs i python 3.5 för att köras med en flask-server på en RaspberryPi 3B med operativsystemet Raspbian.

## 4. Nuvarande design



Det är utifrån denna design som utveckling har skett hitintills. Till en början resonerade vi om att även göra ExternalsController till en egen tråd som skulle köras oberoende av servern men det visade sig att det inte var nödvändigt utan att det funkade ändå.

## 5. Testning

För varje metod skrivs minst ett unit-test som körs med ramverket PyTest. Detta är gjort med alla nuvarande implementerade metoder och ska göras med alla ännu icke implementerade metoder.

## 6. Gränssnitt

Som man kan se designen i under rubrik 3 så är det ett antal publika metoder som andra komponenter förlitar sig på.

- `getInstance()` - statisk metod som returnerar den enda instansen av klassen.
- `readPlantStatus(plantID)` - returnerar plantans nuvarande fuktvärde. Om plantan inte blivit skapad tidigare så skapas den också.
- `createPlant(plantID)` - skapar en ny planta.
- `waterPlant(plantID)` - påbörjar bevattning av plantan.
- `turnOffSprinkler(plantID)` - stänger av bevattning till plantan.
- `updateMinDryness(plantID, minDryness)` - sätter ett nytt börvärde för plantans fuktvärde då den skickar notis till användaren.
- `terminatePlant(plantID)` - stoppar den angivna plantprocessen.
- `isActive(plantID)` - returnerar true eller false beroende på om det finns en aktiv process för den angivna plantan.
- `setEmailForPlant(plantID, userEmail)` - anger till vilken email som notis ska skickas när den angivna plantan är torr.

## 7. Beroenden

Dom beroenden som komponenten har är gentemot plant-modulen i projektet och en av pythons standardmoduler `threading`.

## 8. Intern struktur

Komponentens huvudsakliga uppgift är i botten att hålla reda på dom olika plant-trådarna. Hur detta görs är genom att man har gjort klassen till en singleton så att den enda instansen som finns skapar alla andra plant-trådar och sparar dom i en av pythons standard-datastrukturer, `dict`, som är en slags key-value-map. När någon annan då vill utföra någon operation på en planta så plockar `externalscontroller` då fram den plant-tråden som operationen ska utföras på och utför sedan operationen på den plant-tråden.

## 9. TODO

Just nu finns det inga fler publika metoder som behöver implementeras eller något som är oklart. Om man i framtiden vill ändra på designen så är det fritt fram men då får man se till att skriva nya tester och metoder baserat på dom ändringar som görs, samt uppdatera detta dokument.