**How to Wrap up AI Model as a Service into the SingularityNet**

[1. Create an API for your model or Service using Protobuf 1](#__RefHeading___Toc1875_1035697085)

[what is an API ? 1](#__RefHeading___Toc2202_1035697085)

[what is a Protobuf ? 1](#__RefHeading___Toc2204_1035697085)

[2. Create a GRPC endpoint for your Service 1](#__RefHeading___Toc2206_1035697085)

[3. Compile the Protobuf 1](#__RefHeading___Toc1877_1035697085)

[4. Integrate created and compiled API with SNET-Daemon 2](#__RefHeading___Toc1879_1035697085)

[5. 3](#__RefHeading___Toc1989_1035697085)

[Guidelines 3](#__RefHeading___Toc2393_1035697085)

[Service Integration Example 3](#__RefHeading___Toc1987_1035697085)

[Language Detection 3](#__RefHeading___Toc2209_1035697085)

[Questions 3](#__RefHeading___Toc2211_1035697085)

# 1. Create an API for your model or Service using Protobuf

during our Alpha Version we had using JSONRpc to define our end points

## what is an API ?

## what is a Protobuf ?

Check from References Documents

# 2. Create a GRPC endpoint for your Service

Create a GRPC endpoint for the service endpoint you exposed in the Protobuf using your favoriate languages. Because we have to attach this particular protobuf to the service that we have developed. The implementation details vary from language to language.

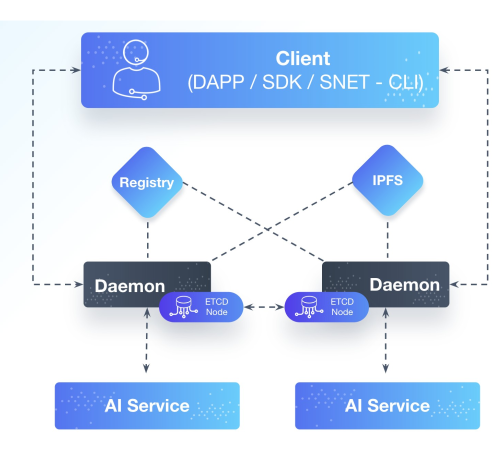
# 3. Compile the Protobuf

* by the language of your choice
* Request & Response parameters to attach to the input and output of the application
* what does it mean by compiling the protobuf file ?
* To install protobuf, we need to install
  + the protocol compiler → used to compile .proto files
  + the protobuf runtime for your choosen language

what does it mean by compling the protobuf

# 4. Integrate created and compiled API with SNET-Daemon

* As we now one can call any grpc endpoint as long as we have the address, It doesn’t have payment logic, authentication or anything that would hamper(track) anyone for using this particular service.
* take a buoutique,if there is no any person who control that boutique, who become the intermediar between the cloth and clients or person, so any person can come and just take the cloth he wants. So to solve this problem we need to have an authorized controler for this boutique who can tell about the service, payments and can receive the payments from clients or person.
* So also Similarly Current AI Service in githubs are just like this boutique without contorler so this is where the daemon comes into place, which is the controller or intermediate between the AI services and Clients(persons)



* Anyone can access the snet service by any of the three Snet Clients
  + DAPP(Decentralised Applications, the web page of beta singularitynet)
  + Snet-Sdk (by using the Sdk)
  + Snet-Cli (you can access the Client commands)
* As we see from the above architecture, the most important thing we need to make sure is the **daemon** end points, because the client communicate with the daemon to access the Service he wants.
* And the daemon contains Registry and IPFS to check about that service and payments, as the controller of the boutique contains a piece of paper that contains about the cloths
* Snet-Daemon, exposes(shows, reveals) an AI applications as an API that is accessible through the SingularityNET Network.
  + It handles all interaction with the block chain for authorization and payment using AGI tokens and passes through API calls to the AI application
* Snet-Daemon is greater than a Person that control a boutique, bc this person controls only small boutique that contains only trousers. Other boutique that contains may be shoe is controlled by another person because one person can not control all this things. But all AI services in SingularityNET marketplace is controlled by one controller which is Snet-Daemon. So any one who develop his own services and wants to make it avialable in the SingularityNET market place has need to first create an API of his service using protobuf file means in the format easily readable by Snet-Daemon and then integrate it with this Snet-Daemon.
* the points is how to write this daemon codes ?

# 5.

# Guidelines

Guidelines that help developers to write/integrate new AI services to the SingularityNET platform

* Supported languages
  + SNET services use gRPC to connect services together or to call a required service in other service e.g french to english translation service can call Text-to-Speech Service through gRPC
  + gRPC is a modern open source high performance RPC framework developed by google
  + RPC(remote procedure call) is a protocol that one program use to request a service from a program located in another computer on a network.
  + So any new service must provide its API in gRPC
  + gRPC Support Several programming languages C++, Java, Python, Go, Ruby, C#, Node.js…
* AI frameworks
* Third party code and models
* Service documentation
* Contributing to existing projects

# Service Integration Example

→ We will Integrate

* Language Detection from the PolyGlot Library
* others might want to use example-service repo

## Language Detection

# Questions