Notebook Research

**Step 1: Create Entity 🡪** Define a class which contains the configuration objects, this class will be of type @dataclass. Save these entities inside config\_entity.py under entities folder.

A screen shot of a computer code

Description automatically generated with low confidence

A screen shot of a computer code

Description automatically generated with low confidence

**Step 2: Create Configuration class 🡪** this class returns configuration of type entity (defined in step 1). This configuration can be used later to retrieve values of configurations. Save these configuration inside configuration.py under config folder.

A picture containing text, screenshot, font, software

Description automatically generated

A screenshot of a computer program

Description automatically generated with medium confidence

**Step 3: Create Action Class (Data Ingestion / Prepare Base Model etc)🡪** this class performs the actual task and used configuration as an input argument. Define this class inside components folder under a separate file (data\_ingestion.py / prepare\_base\_model.py**)**

A picture containing text, screenshot, font

Description automatically generated

A picture containing text, screenshot, font, number

Description automatically generated

**Step 4: Create Pipeline code🡪** this logic puts things in actions by calling everything in sequence. Put this action inside pipeline folder inside a new file (e.g stage\_01\_data\_ingestion.py, stage\_02\_prepare\_base\_model.py). Create a class of the action like (DataIngestionTrainingPipeline/ PrepareBaseModelTrainingPipeline) and define the below steps inside the main function of this class.

E.g

1. Define configuration object [config = ConfigurationManager()]
2. Use this object to get the configuration of entity type [data\_ingestion\_config = config.get\_data\_ingestion\_config()]
3. Define main action class, and pass this entity configuration object [data\_ingestion = DataIngestion(config=data\_ingestion\_config)]
4. Perform action action required [data\_ingestion.download\_file(), data\_ingestion.extract\_zip\_file()]
5. Step 5: Main class 🡪 define main to call multiple pipelines like below.

STAGE\_NAME = "Data Ingestion Stage"  
  
try:  
 logger.info(f">>>>>>> stage {STAGE\_NAME} started <<<<<<")  
 obj = DataIngestionTrainingPipeline()  
 obj.main()  
 logger.info(f">>>>>>> stage {STAGE\_NAME} completed <<<<<\n<<<")  
except Exception as e:  
 logger.exception(e)  
 raise e  
  
  
STAGE\_NAME = "Prepare Base Model"  
  
try:  
 logger.info(f">>>>>>> stage {STAGE\_NAME} started <<<<<<")  
 obj = PrepareBaseModelTrainingPipeline()  
 obj.main()  
 logger.info(f">>>>>>> stage {STAGE\_NAME} completed <<<<<\n<<<")  
except Exception as e:  
 logger.exception(e)  
 raise e