**TypeScript - Interfaces**

An interface is a syntactical contract that an entity should conform to. In other words, an interface defines the syntax that any entity must adhere to.

Interfaces define properties, methods, and events, which are the members of the interface. Interfaces contain only the declaration of the members. It is the responsibility of the deriving class to define the members. It often helps in providing a standard structure that the deriving classes would follow.

Let’s consider an object −

var person = {

FirstName:"Tom",

LastName:"Hanks",

sayHi: ()=>{ return "Hi"}

};

If we consider the signature of the object, it could be −

{

FirstName:string,

LastName:string,

sayHi()=>string

}

To reuse the signature across objects we can define it as an interface.

## Declaring Interfaces

The interface keyword is used to declare an interface. Here is the syntax to declare an interface −

### Syntax

interface interface\_name {

}

### Example: Interface and Objects

interface IPerson {

firstName:string,

lastName:string,

sayHi: ()=>string

}

var customer:IPerson = {

firstName:"Tom",

lastName:"Hanks",

sayHi: ():string =>{return "Hi there"}

}

console.log("Customer Object ")

console.log(customer.firstName)

console.log(customer.lastName)

console.log(customer.sayHi())

var employee:IPerson = {

firstName:"Jim",

lastName:"Blakes",

sayHi: ():string =>{return "Hello!!!"}

}

console.log("Employee Object ")

console.log(employee.firstName);

console.log(employee.lastName);

The example defines an interface. The customer object is of the type IPerson. Hence, it will now be binding on the object to define all properties as specified by the interface.

Another object with following signature, is still considered as IPerson because that object is treated by its size or signature.

On compiling, it will generate following JavaScript code.

//Generated by typescript 1.8.10

var customer = { firstName: "Tom", lastName: "Hanks",

sayHi: function () { return "Hi there"; }

};

console.log("Customer Object ");

console.log(customer.firstName);

console.log(customer.lastName);

console.log(customer.sayHi());

var employee = { firstName: "Jim", lastName: "Blakes",

sayHi: function () { return "Hello!!!"; } };

console.log("Employee Object ");

console.log(employee.firstName);

console.log(employee.lastName);

The output of the above example code is as follows −

Customer object

Tom

Hanks

Hi there

Employee object

Jim

Blakes

Hello!!!