

Language Definitions

Sujit Kumar Chakrabarti
sujitkc@iiitb.ac.in

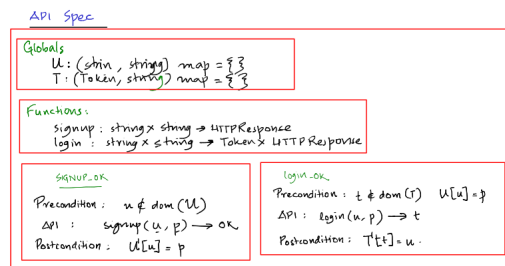
January 2025

1 API Specification Language

1.1 Abstract Syntax

1.2 Example

1.3 Example Specification – Signup-Login API



<i>spec</i>	→	$(g : Decl^*, t : TypeDecl^*, i : Init, f : Function^*, a : API^*)$
<i>Decl</i>	→	$(v : String, t : TypeExpr)$
<i>TypeDecl</i>	→	$VariantDecl \mid RecordDecl$
<i>VariantDecl</i>	→	$VariantConstructor^+$
<i>TypeConstructor</i>	→	$(constrname : String, TypeExpr^*)$
<i>RecordDecl</i>	→	$recname : String, fields : Decl^+$
<i>TypeExpr</i>	→	$TypeConst \mid TypeVariable$ $\mid FuncType$ $\mid MapType$ $\mid TupleType$ $\mid SetType$
<i>TypeConst</i>	→	$String$
<i>FuncType</i>	→	$(args : TypeExpr^*, ret : TypeExpr)$
<i>MapType</i>	→	$(key : TypeExpr, val : TypeExpr)$
<i>TupleType</i>	→	$(elements : TypeExpr^+)$
<i>SetType</i>	→	$(elttype : TypeExpr)$
<i>Init</i>	→	$(v : String, e : Expr)$
Functions		
<i>FunctionDecl</i>	→	$(fname : String, pars : Decl^*, ret : HTTPResponseCode, resp : TypeExpr)$
APIs		
<i>API</i>	→	$(pre : Expr, call : FuncCall, resp : (ret : HTTPResponseCode, resp : post : Expr))$

Figure 1: Abstract Syntax: (a) API Specification Language; (b) Abstract Test Case Language; (c) Common Parts

1.3.1 Abstract Syntax Tree

