Pattern match

Suppose there is a web service return sum of two Int value.

In Python:

def addInt(input):

if "a" in input and "b" in input:

a = input["a"]

b = input["b"]

try:

int(a)

try:

int(b)

return a+b

except:

print b + " is not int"

except:

print a + " is not int"

else:

print "a or b not in input parameters"

While in Scala:

**def** add(requst:String):Option[Int]={  
 **val** inputParas = Json.*parse*(requst)  
 **val** a = (inputParas \ **"a"**).asOpt[Int]  
 **val** b = (inputParas \ **"b"**).asOpt[Int]  
 (a,b) **match** {  
 **case** (a1:Some[Int],b1:Some[Int])=>*Some*(a1.get +b1.get)  
 **case** \_=>None  
 }  
}

In Scala, using pattern match which will only handles the right path of various input types. So we could more focus on the business which matters.