

Le langage Go

Approfondir Go

Table Driven Test

**Certains tests peuvent devenir
rapidement rébarbatifs à écrire**

Je veux tester ma fonction `process()`

“ **Entrée** `X` **=>** **Sortie** `Y`

Entrée `X + 1` **=>** **Sortie** `Y + 1`

etc

”

**En résumé, on veut vérifier que
certaines entrées produisent certaines
sorties**

**Le Table Test simplifie exactement
cela**

Example

```
func inc(i int) int {  
    return i + 1  
}
```

```
func TestInc1(t *testing.T) {  
    v := inc(1)  
    if v != 2 {  
        t.Errorf("Invalid inc(1) result. expected=%v, got=%v", 2, 1)  
    }  
}
```

Si on veut ajouter un 2ème test...

```
func TestInc1(t *testing.T) {  
    v := inc(1)  
    if v != 2 {  
        t.Errorf("Invalid inc(1) result. expected=%v, got=%v", 2, 1)  
    }  
}  
  
func TestInc2(t *testing.T) {  
    v := inc(2)  
    if v != 3 {  
        t.Errorf("Invalid inc(1) result. expected=%v, got=%v", 3, 2)  
    }  
}
```


**Avec un test de table, on peut
factoriser ces tests**

Test Table

```
func TestIncTable(t *testing.T) {  
    var tests = []struct {  
        in int           // argument d'entrée  
        expected int      // résultat attendu  
    } {  
        {1, 2},          // définition des cas  
        {2, 3},  
        {3, 4},  
    }  
    for _, tt := range tests {  
        v := inc(tt.in)  
        if v != tt.expected {  
            t.Errorf("Invalid inc(%v) result. expected=%v, got=%v", tt.in, tt.expected, v)  
        }  
    }  
}
```

Allons écrire un test en code