Cheat Sheet für CSharpFunctionalExtensions

Name	Beschreibung	Hinweis	Beispiel
.Map()	Nimmt das Rückgabeobjekt und packt es in ein Result.	Ist das Rückgabeobjekt ein Result, so erhält man ein Result in einem Result.	result.Map(() => "o_O");
.MapTry()	Wie .Map(), aber mit internem Try/Catch.	-	-
.Bind()	Ersetzt das aktuelle Result mit dem Rückgaberesult.	Rückgabe muss zwingend ein Result oder Result<T> sein.	result.Bind(() => Resul.Success("o_O"));
.BindTry()	Wie .Bind(), aber mit internem Try/Catch.	-	-
.Tap()	Führt den übergebenen Code aus, ohne das Result zu verändern.	Rückgabeobjekte werden ignoriert.	result.Tap(x => Print(x));
.TapTry()	Wie .Tap(), aber mit internem Try/Catch.	-	-
.Check()	Prüft, ob ein Methodenaufruf ein Success-Result liefert.	Rückgabe muss zwingend ein Result oder Result<T> sein.	result.Check(i => Validate(i));
.Ensure()	Prüft gegen ein mitgegebens Predikat und setzt im Fehlerfall den Error.	Das aktuelle Objekt im Result kann geprüft werden.	result.Ensure(i => i>0, "i muss größer 0 sein");
.Compensate()	Übergebene Function wird ausgeführt, wenn aktuelles Result ein Failure ist.	Wie .Bind(), aber nur im Fehlerfall.	
.Finally()	Wird in jedem Fall ausgeführt und lässt Zugriff auf eigentliches Result zu.	Gibt das Rückgabeobjekt direkt nach aussen weiter.	result.Finally(result => result.Value);
.ConvertFailure<T>()			
Result .Combine()	Kombiniert eine Liste von Results zu einem zentralen Result (ohne Value).	Ideal für Validierungen oder Batch-Operationen.	Result.Combine("\n", ValidateA(a), ValidateB(b));