

1.	Record Nr.	UNINA990005019340403321
	Autore	Capuana, Luigi <1839–1915>
	Titolo	Capuana inedito / [a cura di] Alfredo Barbina
	Pubbl/distr/stampa	Bergamo : Minerva Italica, 1974
	Descrizione fisica	212 p. ; 21 cm
	Disciplina	858.9
	Locazione	FLFBC
	Collocazione	ALPHA 4368
	Lingua di pubblicazione	Italiano
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
2.	Record Nr.	UNINA9910797432103321
	Autore	Fehre Philipp
	Titolo	JavaScript domain-driven design : speed up your application development by leveraging the patterns of domain-driven design / / Philipp Fehre
	Pubbl/distr/stampa	Birmingham, [England] ; ; Mumbai, [India] : , : Packt Publishing, , 2015 ©2015
	ISBN	1-78439-114-X
	Edizione	[1st edition]
	Descrizione fisica	1 online resource (207 p.)
	Collana	Community Experience Distilled
	Disciplina	005.2 005.2762
	Soggetti	Java (Computer program language) Application program interfaces (Computer software)
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Description based upon print version of record.
	Nota di bibliografia	Includes bibliographical references at the end of each chapters and index.

Cover; Copyright; Credits; About the Author; About the Reviewers; www.PacktPub.com; Table of Contents; Preface; Chapter 1: A Typical JavaScript Project; The core idea of domain-driven design; Managing an orc dungeon; Inside the dungeon; Managing incoming prisoners; The current state of the art; Digital dungeon management; Specification; Tracking available cells; Starting outgoing transfers; Tracking the state of incoming transfers; Initiating incoming transfers; From greenfield to application; The first route and model; Creating the model; The first route and loading the dungeon
Displaying the pageGluing the application together via express; Moving the application forward; Another look at the problem; Thinking in an MVC web application; Understanding the core problem; Communication is key; The concepts of domain-driven design; It is all about distractions; Focus on the problem at hand; Further reading; Summary; Chapter 2: Finding the Core Problem; Exploring a problem; Outlining the problem; Tracking knowledge; The medium; Paper programming; So how does such a paper program work?; Not so scary UML; Involving the experts; Finding the gaps; Talking business
Talking about the actorsIdentifying the hard problem; Mapping the dependencies; Drawing with code - spiking; Getting started, it's about time; Creating value without creating code; Deciding on the first feature; Summary; Chapter 3: Setting Up a Project for Domain-driven Design; Structuring a project as we see it; Approachability; Locality of edits; Fitness; Dealing with shared functionality; A shared toolbox; Moving up the dependencies; Testing; Setting up a test environment; Different types of tests and goals; Feature specs; Unit tests; Performance tests; Continuous integration
Managing the buildWhy every application needs a build system; Running the tests; Packaging the application; Deploying; Choosing the right system; Isolating the domain; The architecture of modern applications; Hexagonal architecture; Applying the pattern; Plugging in a framework; Summary; Chapter 4: Modelling the Actors; The shoulders of giants; The Different approaches to development; Introducing mocks; Why and why not to mock; Who is involved in the prisoner transfer?; Different objects and their roles; Naming objects according to the domain; The traps of common names like *Manager
Readability of method namesObjects first; The basics of objects in JavaScript; Inheritance and why you won't need it; Modeling patterns beyond inheritance; The object compositionomposition; Polymorphism without inheritance; Applying object design to the domain; Building a system on simple objects; Summary; Chapter 5: Classification and Implementation; Building a common language; The importance of object classification; Seeing the bigger picture; Value objects; The advantages of value objects; The referential transparency; Objects defined as entities; More on entities
Managing the application's lifecycle

Sommario/riassunto

If you are an experienced JavaScript developer who wants to improve the design of his or her applications, or find yourself in a situation to implement an application in an unfamiliar domain, this book is for you. Prior knowledge of JavaScript is required and prior experience with Node.js will also be helpful.
